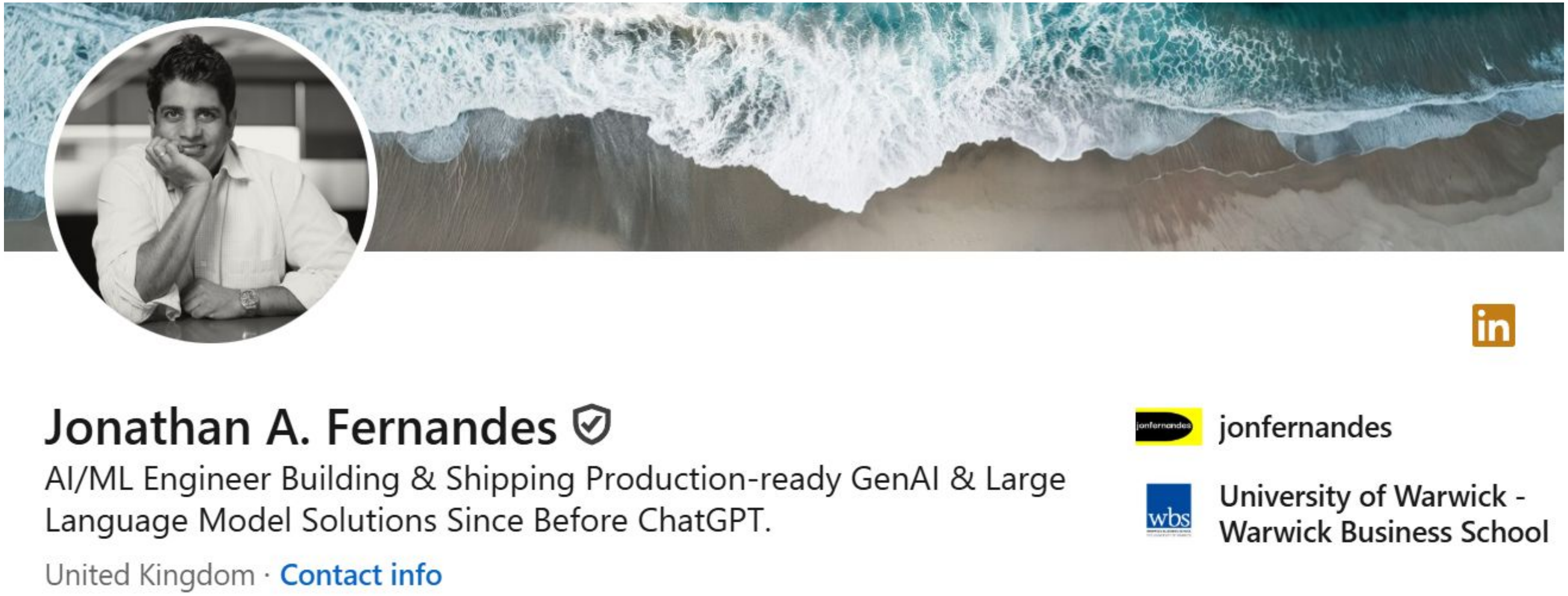
O'REILLY®

# Fundamentals of Large Language Models

# About me



Jonathan A. Fernandes

AI/ML Engineer Building & Shipping Production-ready GenAI & Large Language Model Solutions Since Before ChatGPT.

United Kingdom · Contact info

jonfernandes

University of Warwick - Warwick Business School

# Things you need for today

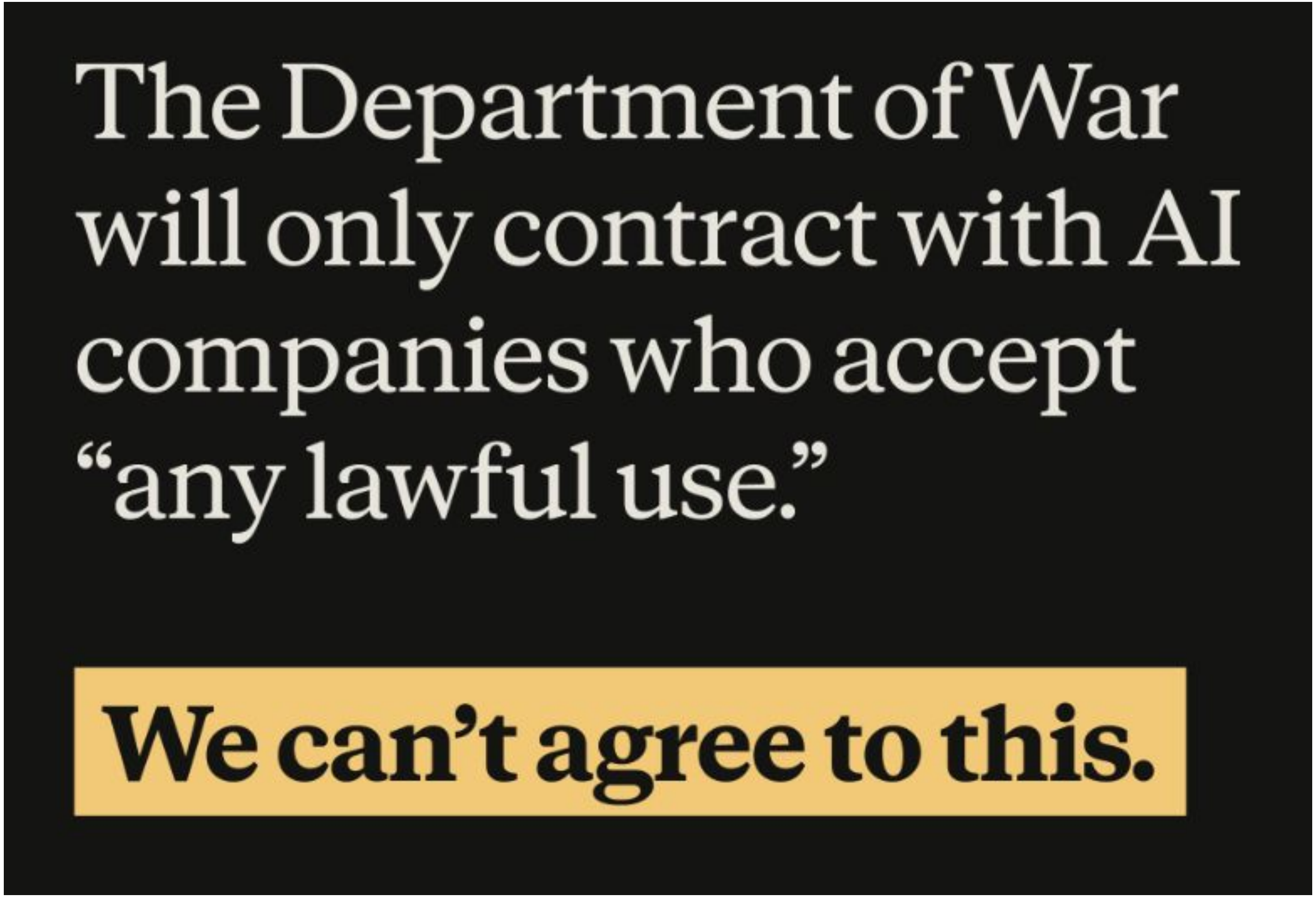
- OpenAI account - https://platform.openai.com/playground
- Cohere account - https://dashboard.cohere.ai/playground

What questions about Large Language Models would you like covered today?

Please put this in the Q&A

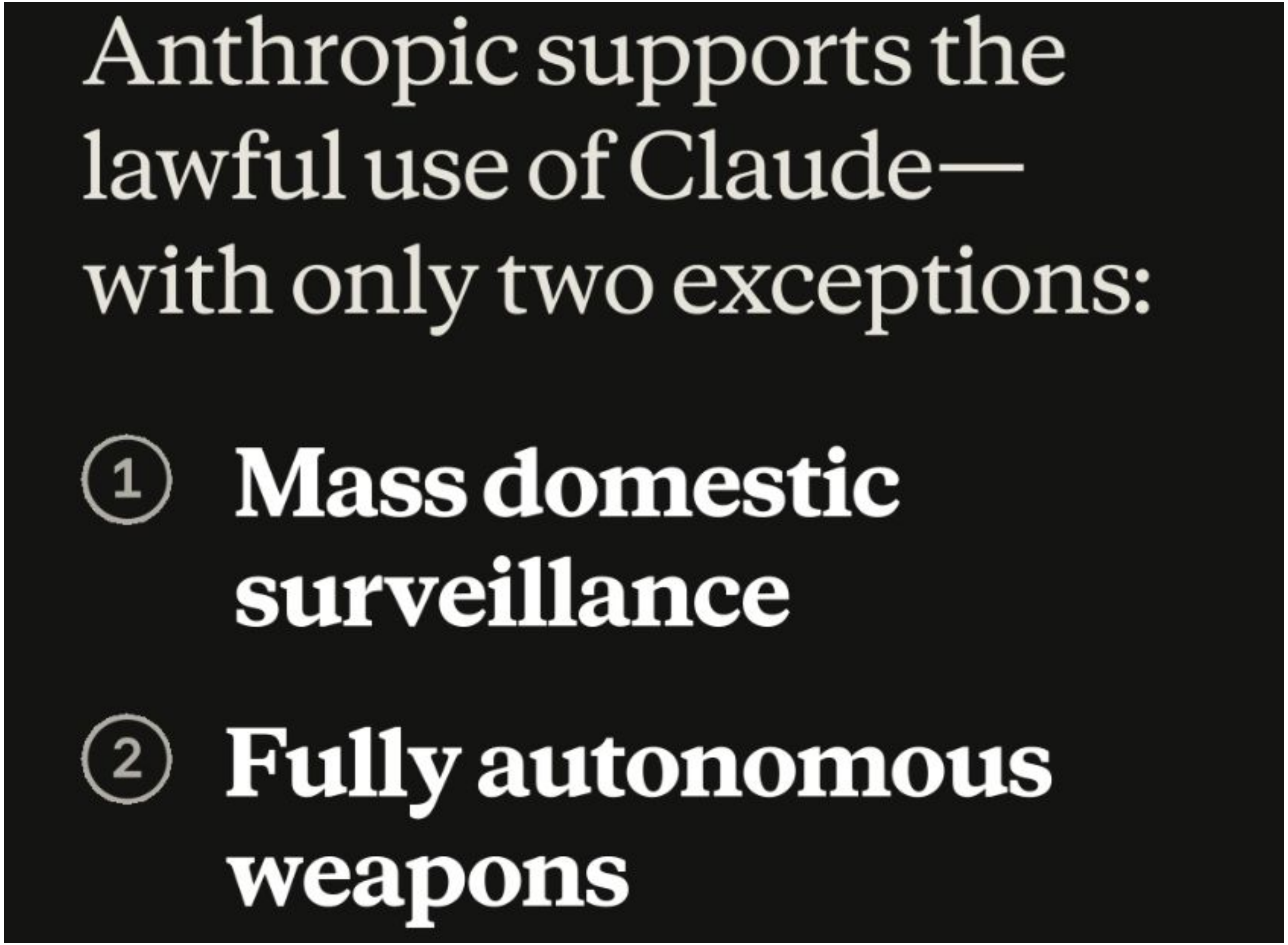
This online training is always being updated.

# Anthropic and the Department of War



The Department of War will only contract with AI companies who accept "any lawful use."

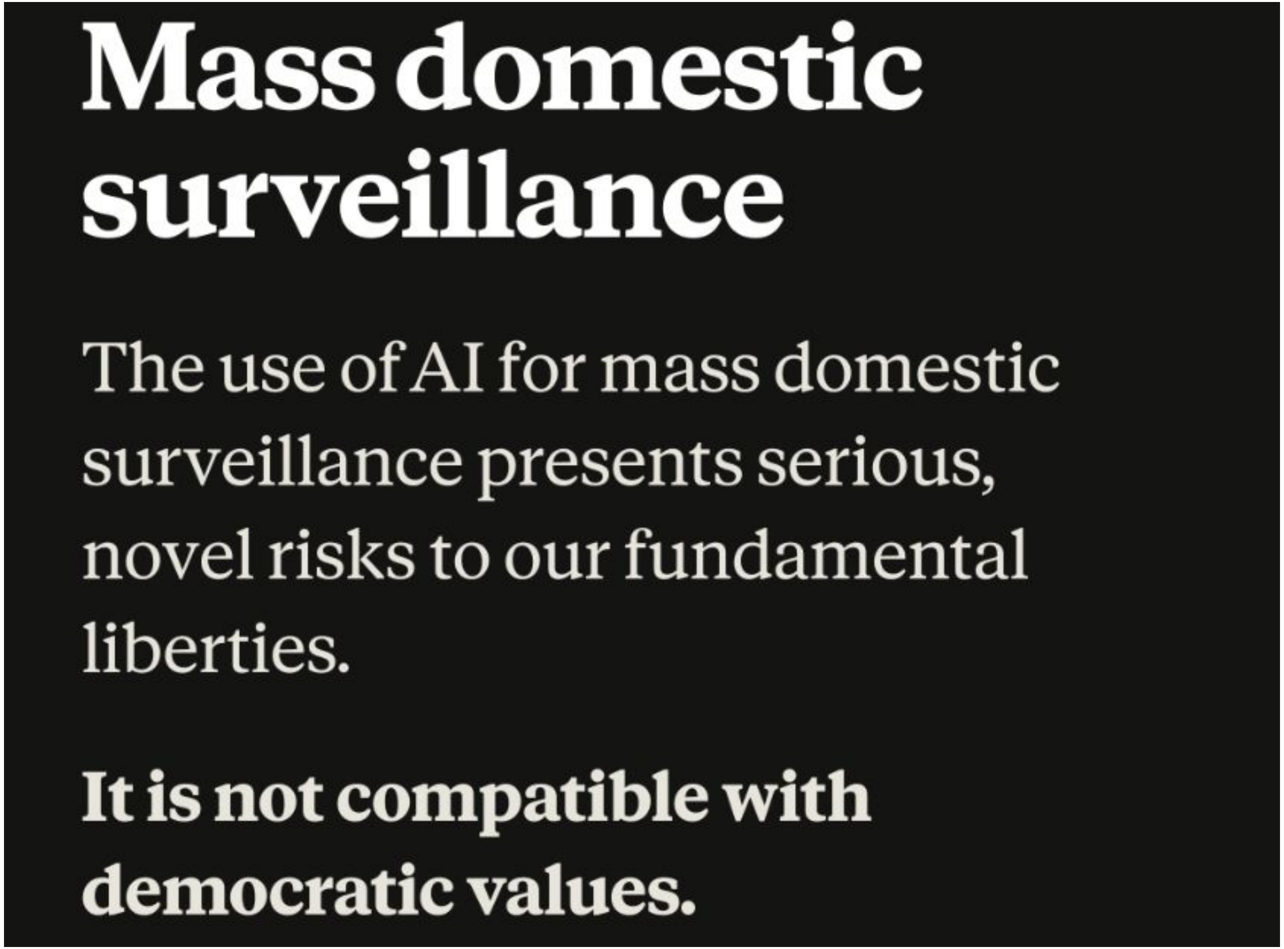
**We can't agree to this.**



Anthropic supports the lawful use of Claude—with only two exceptions:

① **Mass domestic surveillance**

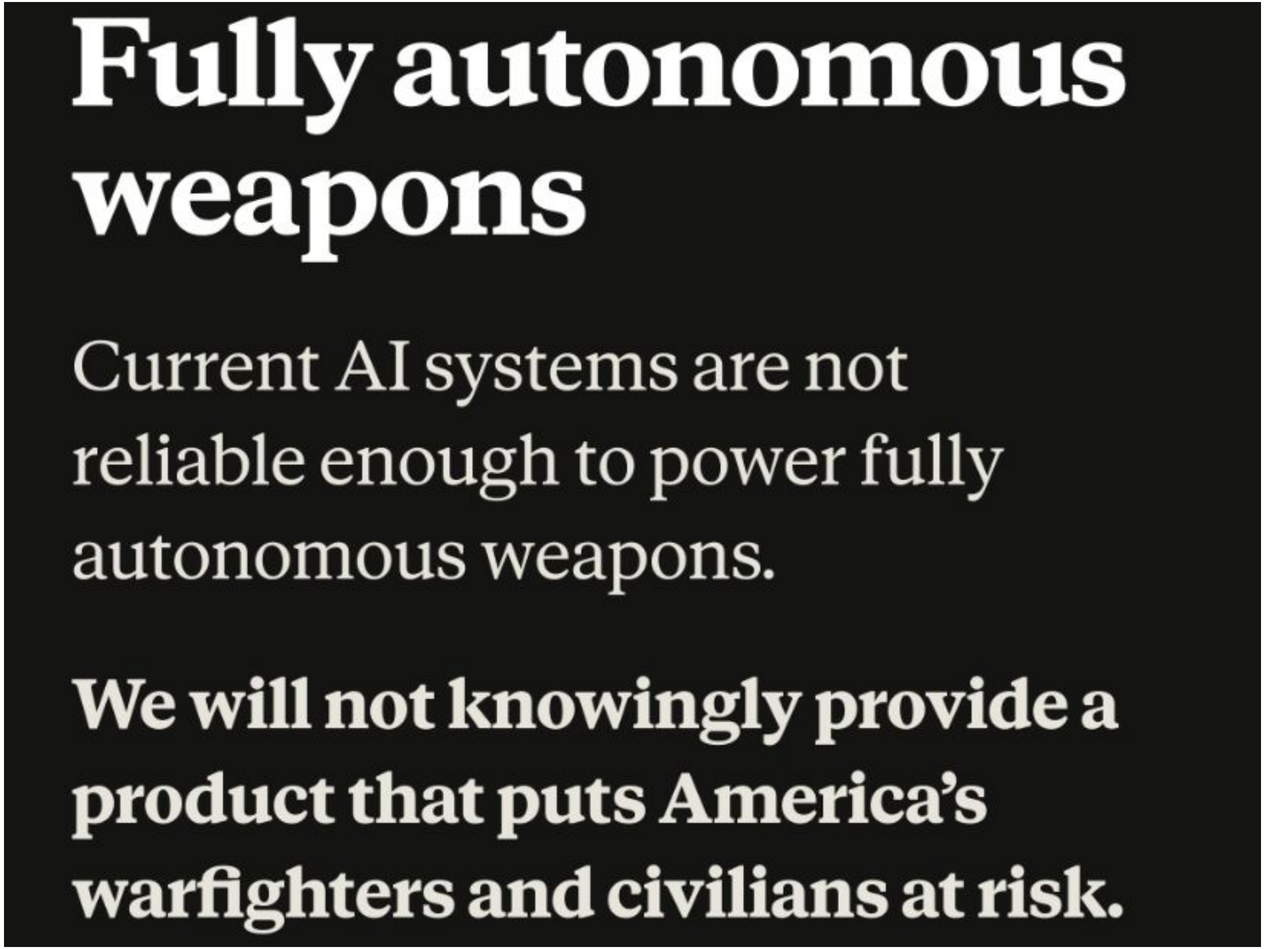
② **Fully autonomous weapons**



# Mass domestic surveillance

The use of AI for mass domestic surveillance presents serious, novel risks to our fundamental liberties.

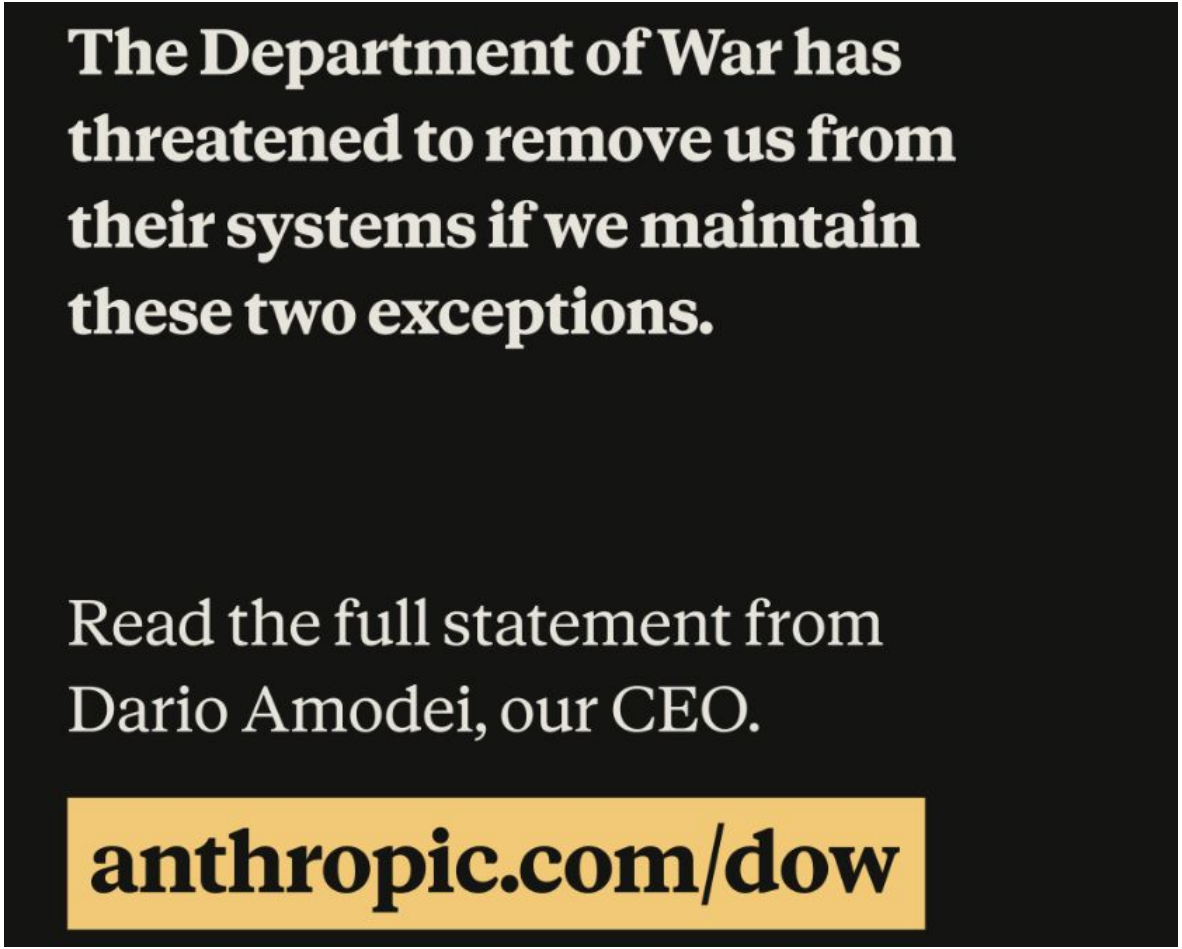
**It is not compatible with democratic values.**



## Fully autonomous weapons

Current AI systems are not reliable enough to power fully autonomous weapons.

**We will not knowingly provide a product that puts America's warfighters and civilians at risk.**



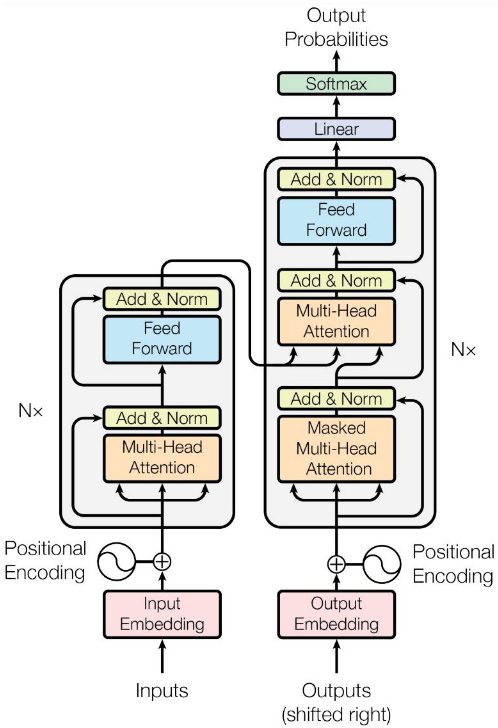
**The Department of War has threatened to remove us from their systems if we maintain these two exceptions.**

Read the full statement from Dario Amodei, our CEO.

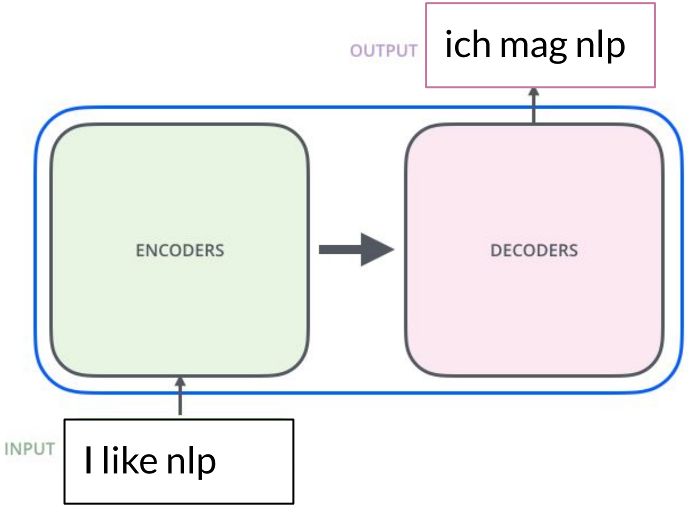
**anthropic.com/dow**

# Transformer: Architecture Overview

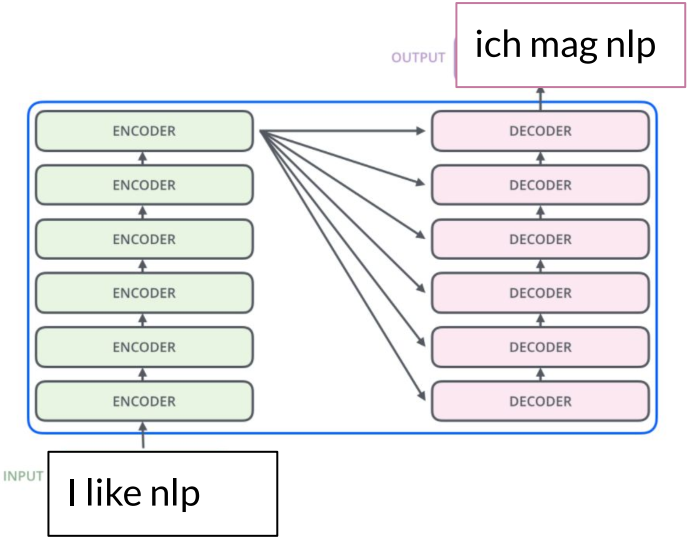
# Transformer architecture



# Transformer overview

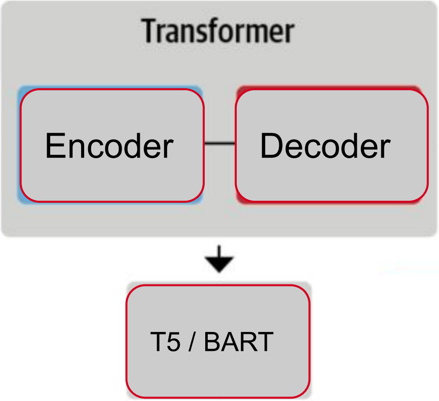


# Transformer overview



# Encoder-decoder model

- Generative tasks



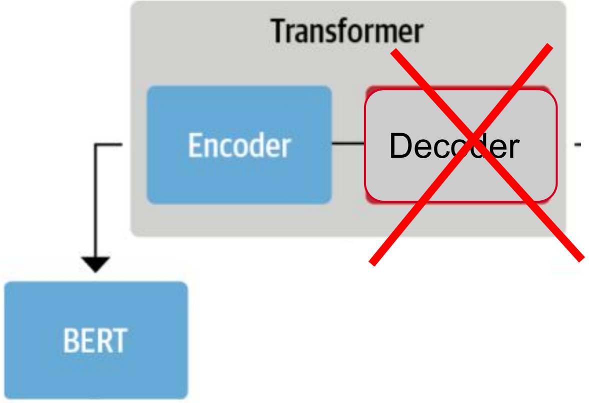
# Encoder-only model

Understanding of input

- Sentence classification
- Named Entity Recognition

Family of BERT models:

- BERT, RoBERTa, DistilBERT …

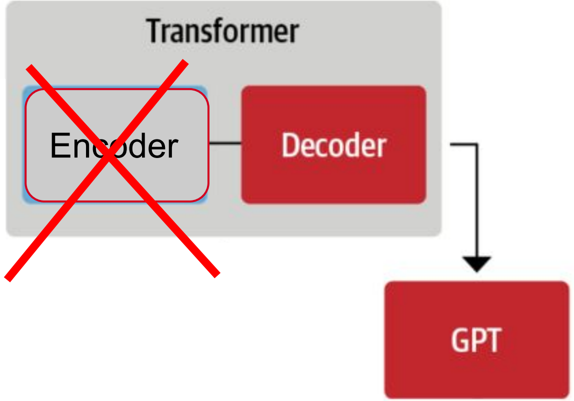


# Decoder-only model

- Generative tasks

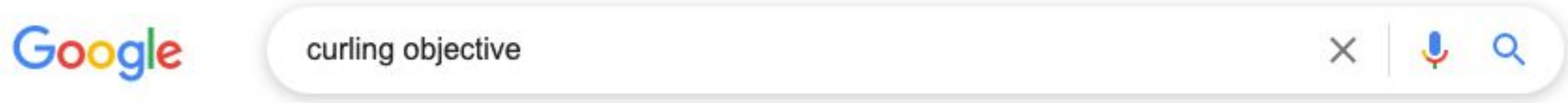
Examples:

- OpenAI GPT models, Claude, Gemini



# Encoder models

ore on
channel.com
LAUSANNE
2020

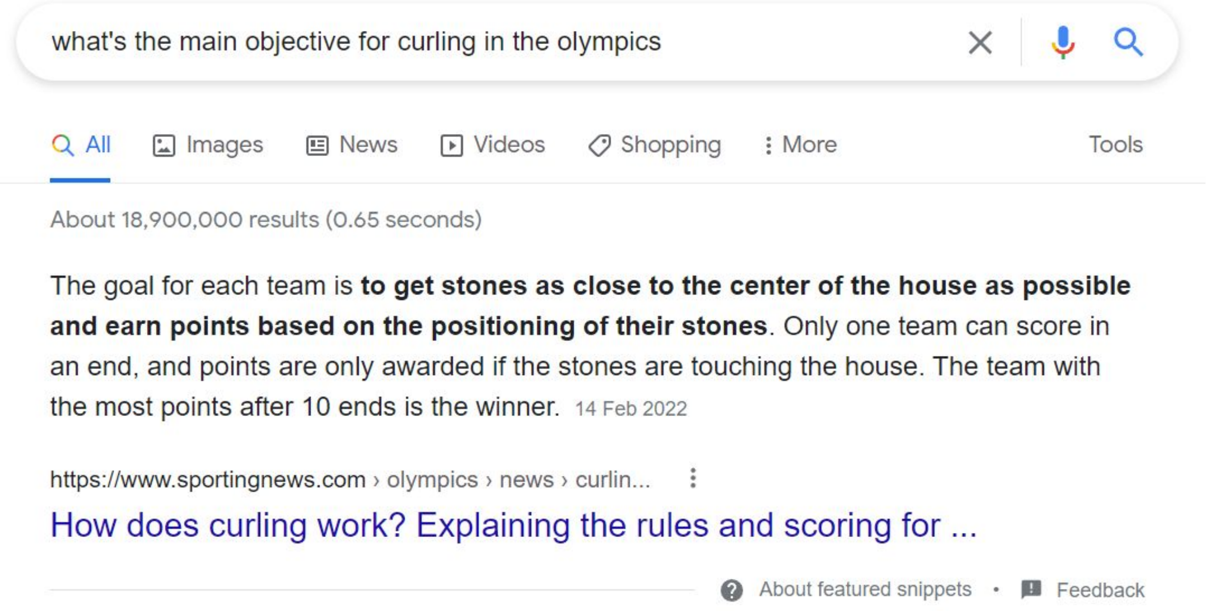


curling objective

# BERT

Bidirectional Encoder Representations from Transformers

# Where are Transformers used in production?



what's the main objective for curling in the olympics

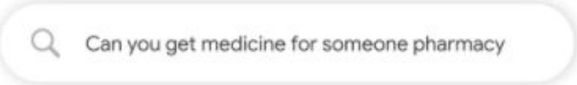
About 18,900,000 results (0.65 seconds)

The goal for each team is **to get stones as close to the center of the house as possible and earn points based on the positioning of their stones**. Only one team can score in an end, and points are only awarded if the stones are touching the house. The team with the most points after 10 ends is the winner. 14 Feb 2022

https://www.sportingnews.com › olympics › news › curlin...

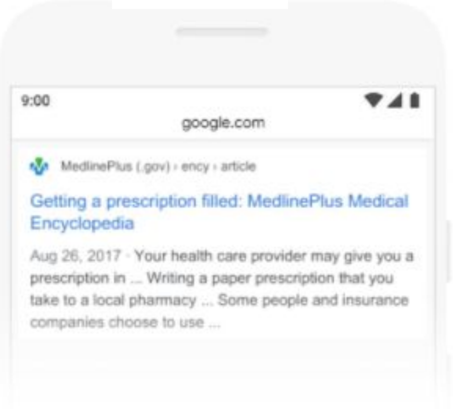
How does curling work? Explaining the rules and scoring for ...

# Transformers in production



Can you get medicine for someone pharmacy

BEFORE BERT



9:00

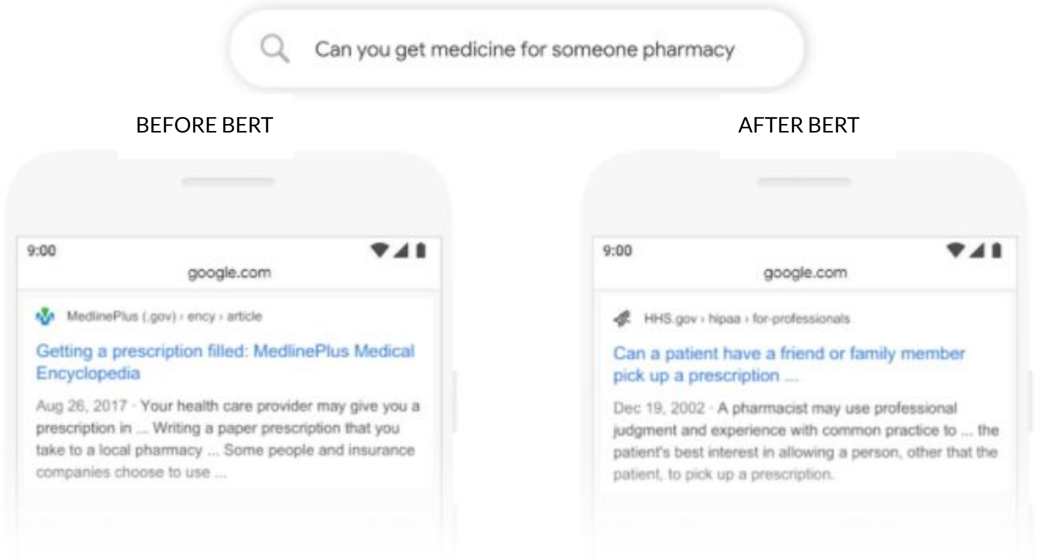
google.com

MedlinePlus (.gov) › ency › article

Getting a prescription filled: MedlinePlus Medical Encyclopedia

Aug 26, 2017 · Your health care provider may give you a prescription in ... Writing a paper prescription that you take to a local pharmacy ... Some people and insurance companies choose to use ...

# Transformers in production



Can you get medicine for someone pharmacy

BEFORE BERT

9:00

google.com

MedlinePlus (.gov) › ency › article

Getting a prescription filled: MedlinePlus Medical Encyclopedia

Aug 26, 2017 · Your health care provider may give you a prescription in ... Writing a paper prescription that you take to a local pharmacy ... Some people and insurance companies choose to use ...

AFTER BERT

9:00

google.com

HHS.gov › hipaa › for-professionals

Can a patient have a friend or family member pick up a prescription ...

Dec 19, 2002 · A pharmacist may use professional judgment and experience with common practice to ... the patient's best interest in allowing a person, other that the patient, to pick up a prescription.

# What was BERT trained on?

BERT - Wikipedia and BooksCorpus (11,000 unpublished books)

# What tasks was BERT trained?

- Masked Language Model (MLM)
- Next Sentence Prediction (NSP)

The Tokyo Olympic games were  <masked> from 2020 to 2021.

# Masked Language Modelling (MLM)

The Tokyo Olympic games were  <masked> from 2020 to 2021.

# Masked Language Modelling (MLM)

The Tokyo Olympic games were  postponed from 2020 to 2021.

# Next sentence prediction (NSP)

The Tokyo Olympic games were  postponed from 2020 to 2021. This is the first instance in the history of the Olympics as previous games had been cancelled but not rescheduled.

# Why MLM and NSP?

BERT gets a good understanding of English language.

# Pre-training: BERT

| | BERT |
|---|---|
| Year | 2018 |
| Number of parameters | 109M |
| Training time | 12 days |
| Infrastructure | 8 x V100 GPUs (*) |
| Size of dataset used for training | 16GB |
| Training tokens (dataset) | 250B |
| Dataset source | Wikipedia |
| | Book corpus |
| | |
| | |
| | |

# What are tokens?

1500 words is approximately equivalent to 2400 tokens

# What are tokens?

1500 words is approximately equivalent to 2400 tokens

A word is approximately 1.4 tokens

# What are tokens?

1500 words is approximately equivalent to 2400 tokens

A word is approximately 1.4 tokens

A novel is 100,000 words, or 140,000 tokens

# What are tokens?

BERT was trained on 250B tokens or:

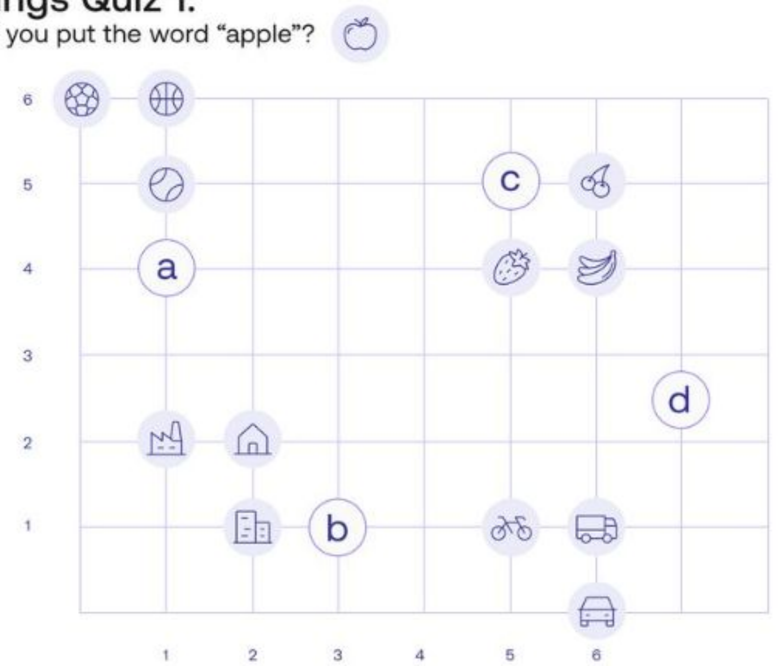
1.8 million novels

# Embeddings

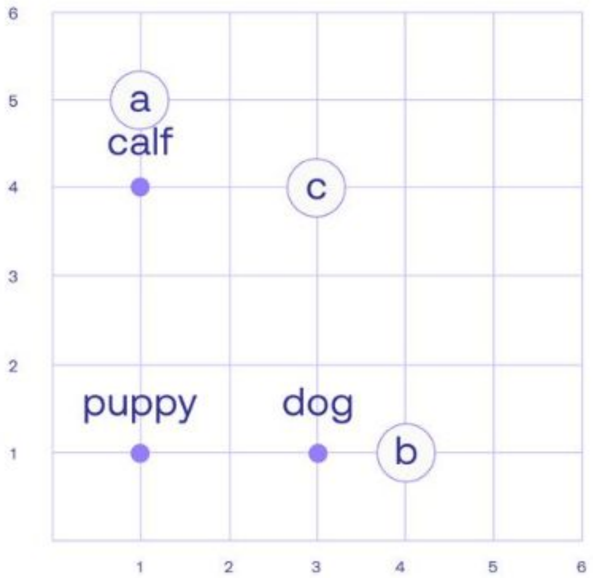
Banana
Basketball
Bicycle
Building
Car
Castle
Cherry
House
Soccer
Strawberry
Tennis
Truck

## Embeddings Quiz 1:

Where would you put the word “apple”?



# What is c?



## Word embeddings

Many more columns

| Word | Numbers | |
|---|---|---|
| Apple | 5 | 5 |
| Soccer | 0 | 6 |
| House | 2 | 2 |
| Car | 6 | 0 |

| Word | Numbers | | | |
|---|---|---|---|---|
| A | -0.82 | -0.32 | ... | -0.23 |
| Aardvark | 0.419 | 1.28 | ... | -0.06 |
| ... | | | ... | |
| Zygote | -0.74 | -1.02 | ... | 1.35 |

4096

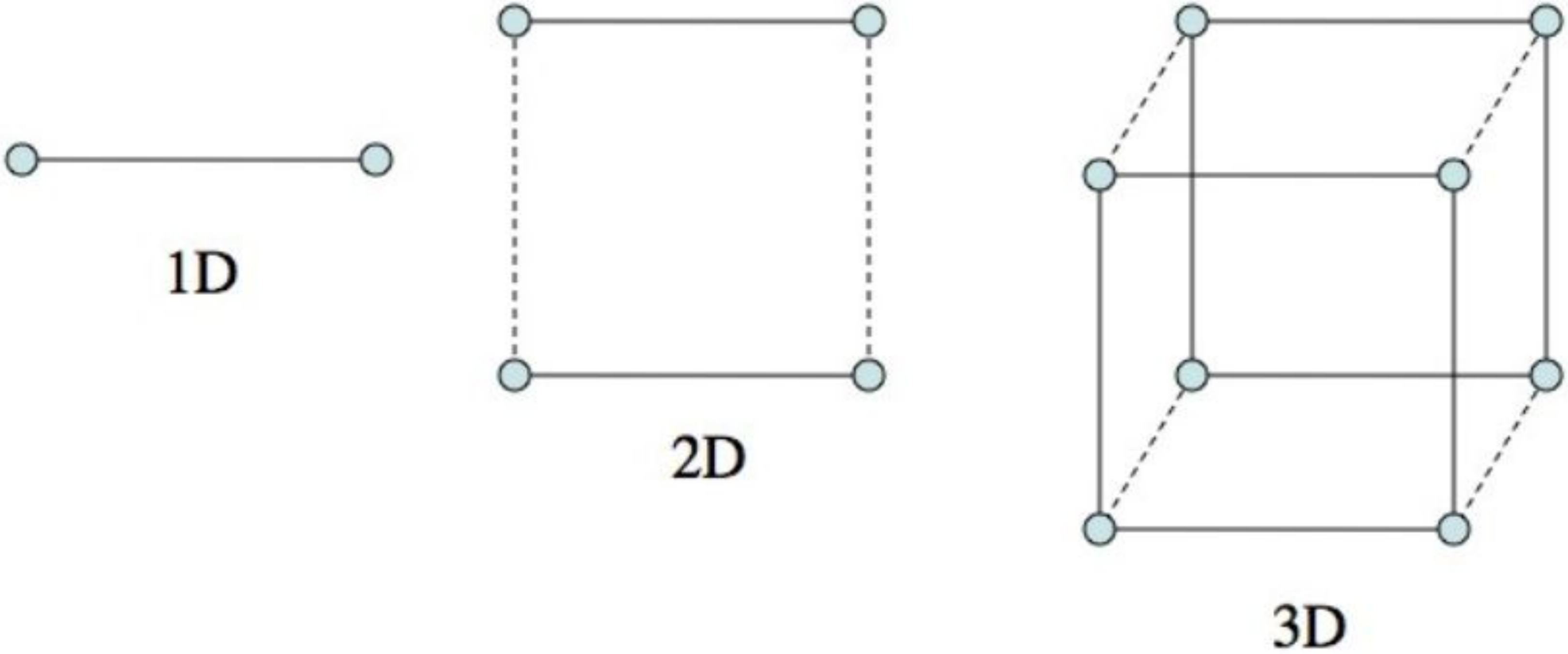
## Sentence embeddings with Cohere (demo)

https://docs.google.com/spreadsheets/d/17AVE0M1mLgOVR1ptDUzP218rVrXbTTzwaQkxDpQlPIQ/edit?usp=sharing

# Similarity between text

- Dot Product
- Cosine Similarity

The more similar two words or sentences are, the larger their Dot Product


1D
2D
3D

Cohere’s embeddings have 4096 dimensions

# What do each of the dimensions mean?

| | Dimension 0 (How citric?) | Dimension 1 (How large?) |
|---|---|---|
| Lemons are rich in vitamin C | 8 | 2 |
| Limes are tangy and acidic | 9 | 1 |
| Michael Jordan played for the Chicago Bulls | 0 | 10 |

# What do each of the dimensions mean?

| | Dimension 0 (How citric?) | Dimension 1 (How large?) |
|---|---|---|
| Lemons are rich in vitamin C | 8 | 2 |
| Limes are tangy and acidic | 9 | 1 |
| Michael Jordan played for the Chicago Bulls | 0 | 10 |

Dot-product between Lemons and Jordan sentence : 8 x 0 + 2 x 10 = 20

# What do each of the dimensions mean?

| | Dimension 0 (How citric?) | Dimension 1 (How large?) |
|---|---|---|
| Lemons are rich in vitamin C | 8 | 2 |
| Limes are tangy and acidic | 9 | 1 |
| Michael Jordan played for the Chicago Bulls | 0 | 10 |

Dot-product between Limes and Jordan sentence : 9 x 0 + 1 x 10 = 10

# What do each of the dimensions mean?

| | Dimension 0 (How citric?) | Dimension 1 (How large?) |
|---|---|---|
| Lemons are rich in vitamin C | 8 | 2 |
| Limes are tangy and acidic | 9 | 1 |
| Michael Jordan played for the Chicago Bulls | 0 | 10 |

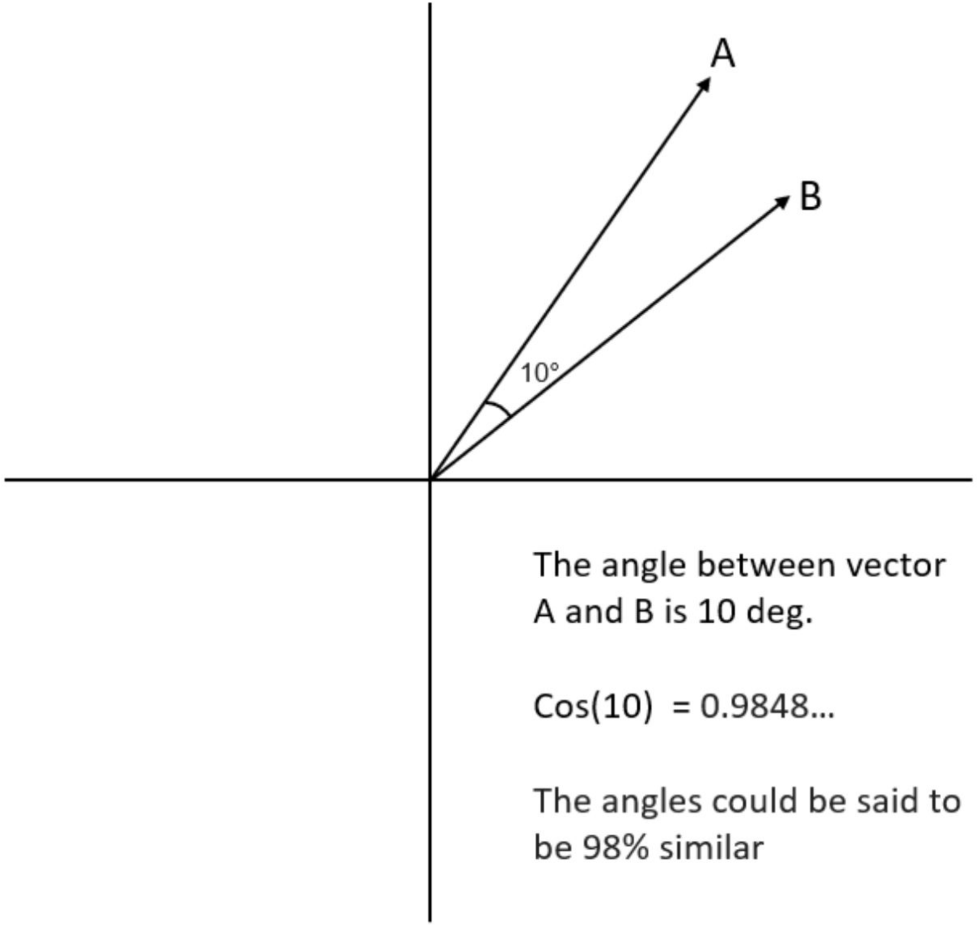
Dot-product between Limes and Lemons sentence : 8 x 9 + 2 x 1 = 74

Can we have a similarity score between 0 and 1?

Cosine Similarity:

- 2 sentences that are very dissimilar have a score close to 0.
- 2 sentences that are similar have a score close to 1.

# Cosine Similarity



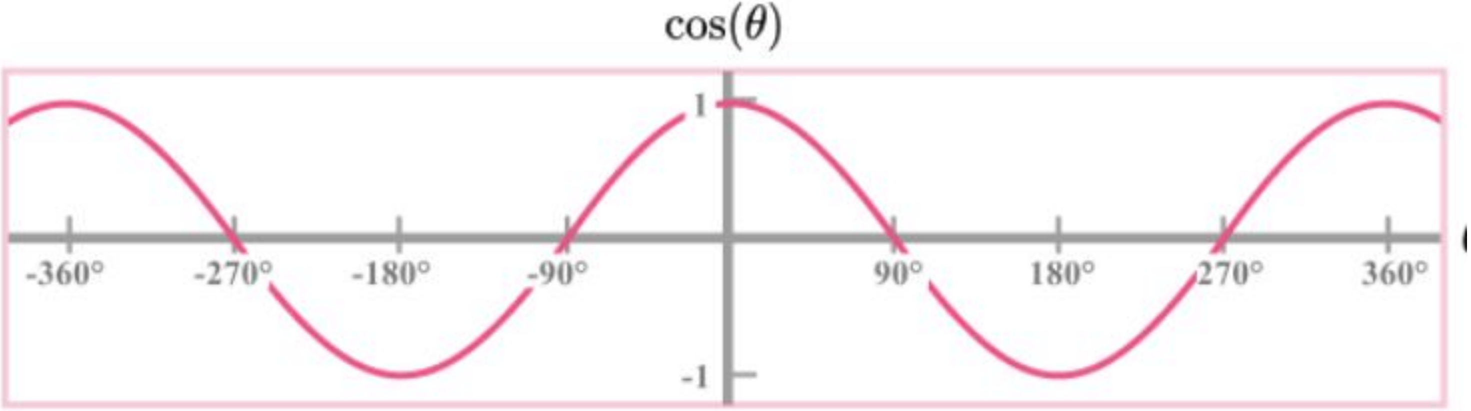
The angle between vector A and B is 10 deg.

Cos(10) = 0.9848...

The angles could be said to be 98% similar

# Cosine Similarity:

- 2 sentences that are very dissimilar have a score close to 0.
- 2 sentences that are similar have a score close to 1.



**Colab notebook (7 minutes):**
https://colab.research.google.com/drive/1YVy0zrz42z2WexDYUFHMu9XMRluJgKB5

# Multilingual embedding models

# Multilingual demo

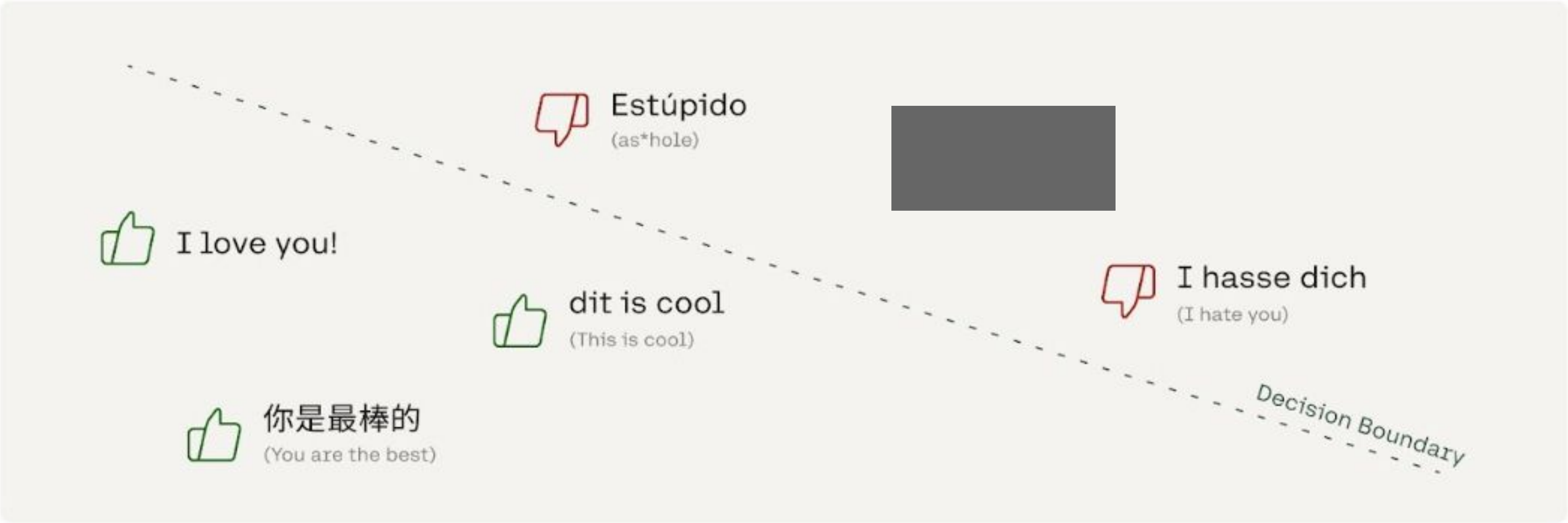
https://docs.google.com/spreadsheets/d/11alaXzWwwVk9U8mVjFbGGkoqNzxWBuAGF6tVjB3O_T8/edit?usp=sharing

# What are some  applications for multilingual embeddings?

# What are some applications for multilingual embeddings?

- **Sentiment Analysis:** Analyze customer sentiment in any language.
- **Content Moderation:** Tackle spam and hate-speech in international communities like online gaming.
- **Intent Recognition:** Classify the user's intent based on a set of predefined intents (e.g., booking a flight, ordering food, etc.).

# Cross-lingual classification



Estúpido
(as*hole)

[redacted]

I love you!

dit is cool
(This is cool)

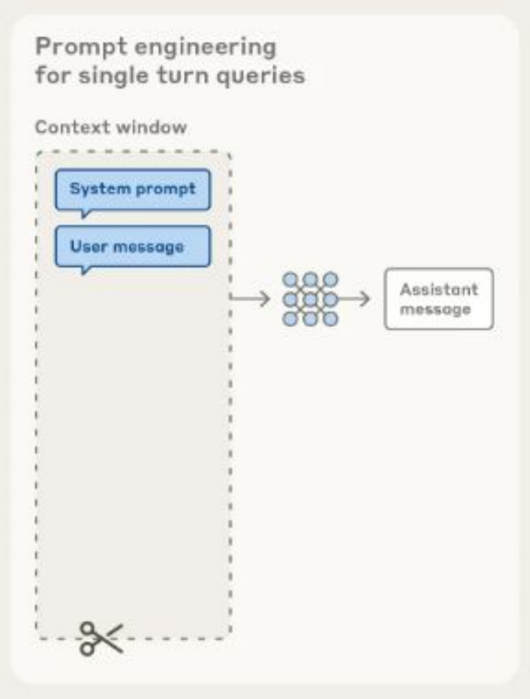
I hasse dich
(I hate you)

你是最棒的
(You are the best)

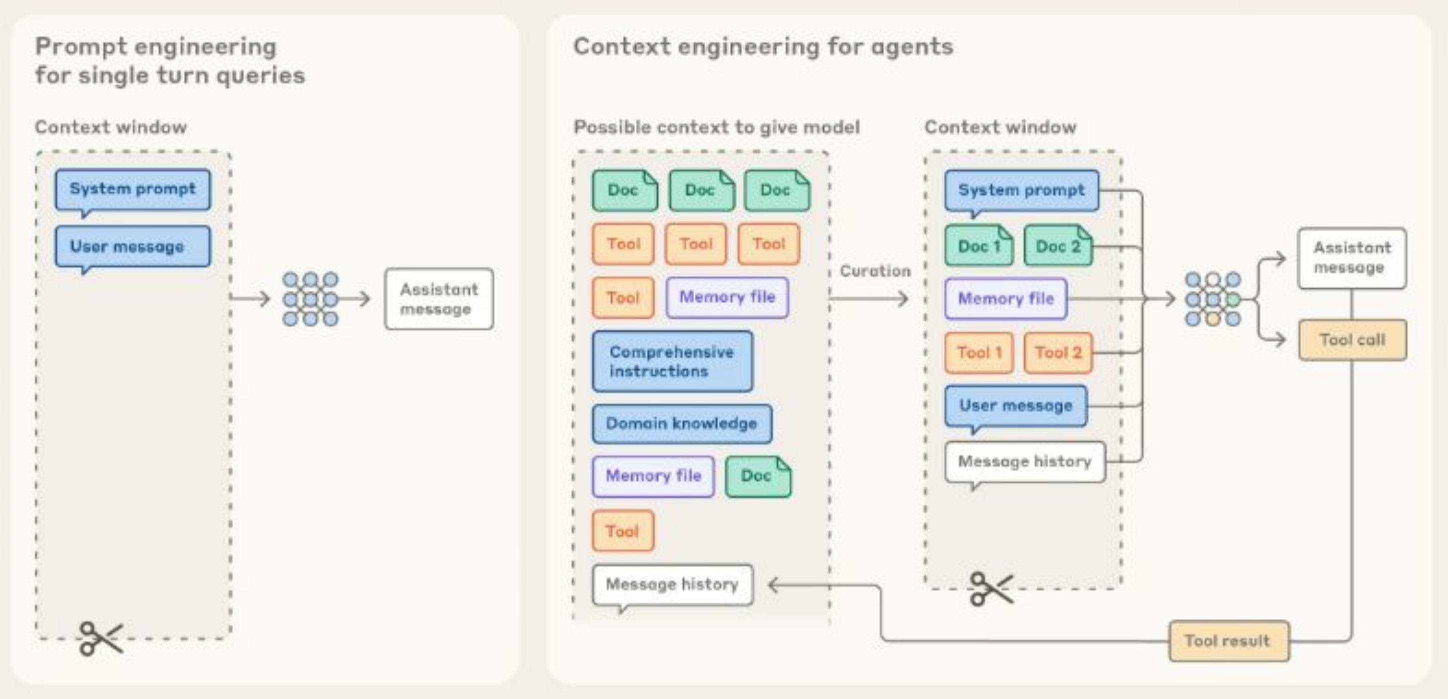
Decision Boundary

# Context Engineering

# Difference between Prompt Engineering & Context Engineering

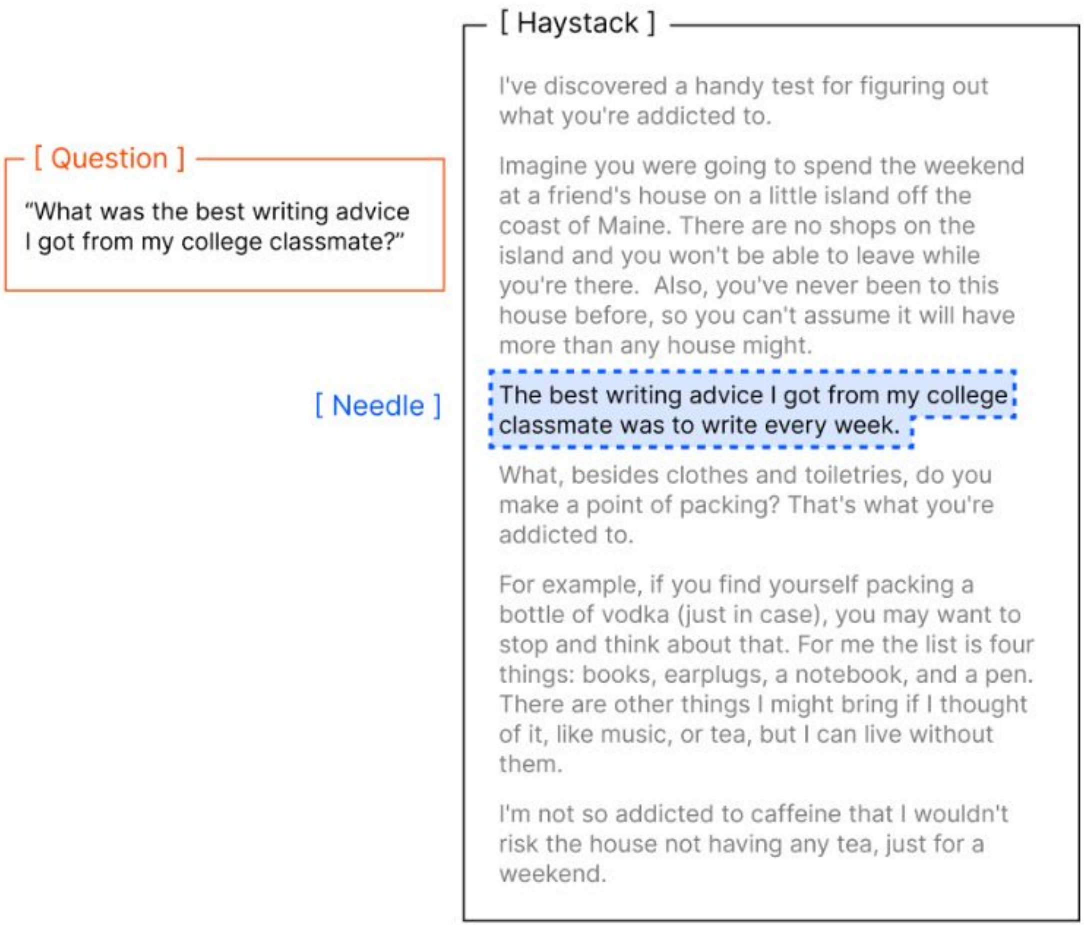


# Difference between Prompt Engineering & Context Engineering



# Working with context constraints

Context length isn't context depth; LLMs don't actually process the 10,000th token as reliably as the first.



[ Question ]

“What was the best writing advice I got from my college classmate?”

[ Needle ]

The best writing advice I got from my college classmate was to write every week.

[ Haystack ]

I've discovered a handy test for figuring out what you're addicted to.

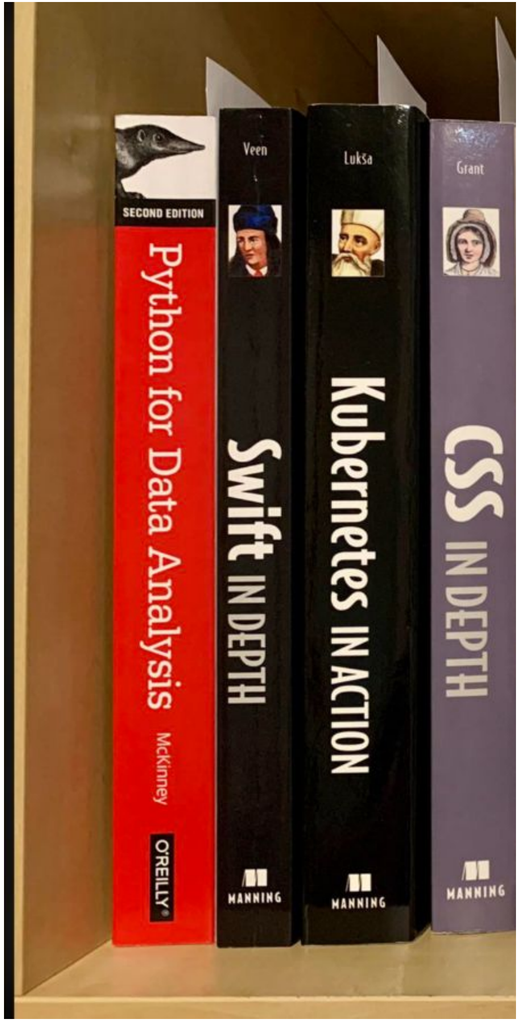
Imagine you were going to spend the weekend at a friend's house on a little island off the coast of Maine. There are no shops on the island and you won't be able to leave while you're there.  Also, you've never been to this house before, so you can't assume it will have more than any house might.

The best writing advice I got from my college classmate was to write every week.

What, besides clothes and toiletries, do you make a point of packing? That's what you're addicted to.

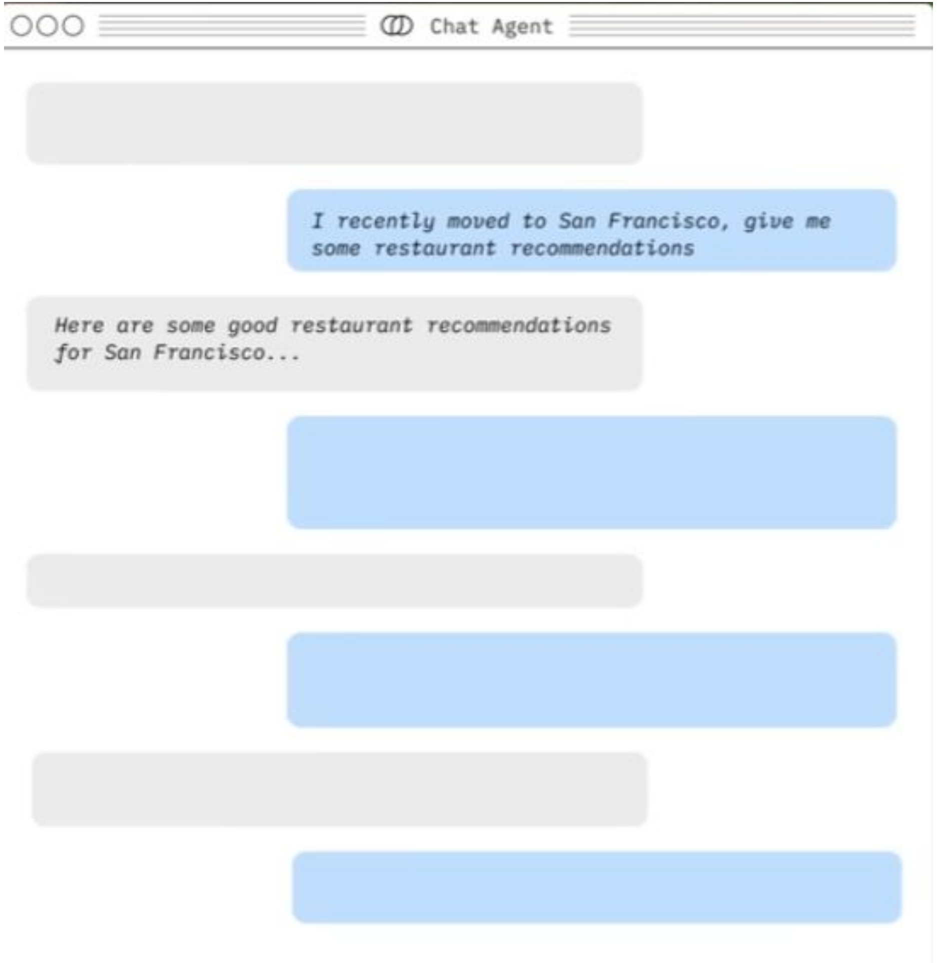
For example, if you find yourself packing a bottle of vodka (just in case), you may want to stop and think about that. For me the list is four things: books, earplugs, a notebook, and a pen. There are other things I might bring if I thought of it, like music, or tea, but I can live without them.

I'm not so addicted to caffeine that I wouldn't risk the house not having any tea, just for a weekend.



SECOND EDITION
Python for Data Analysis
McKinney
O'REILLY
Veen
Swift IN DEPTH
MANNING
Lukša
Kubernetes IN ACTION
MANNING
Grant
CSS IN DEPTH
MANNING

Chat Agent

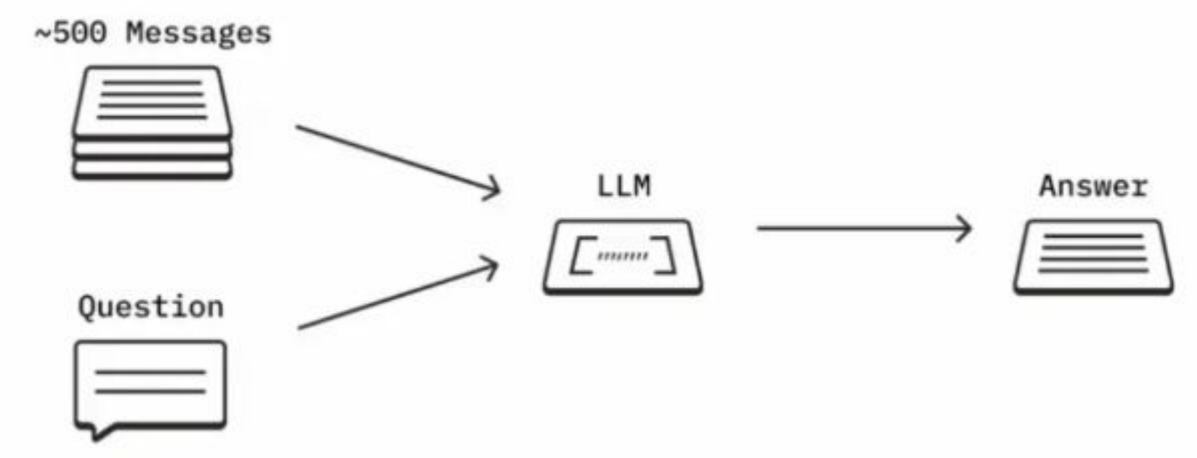


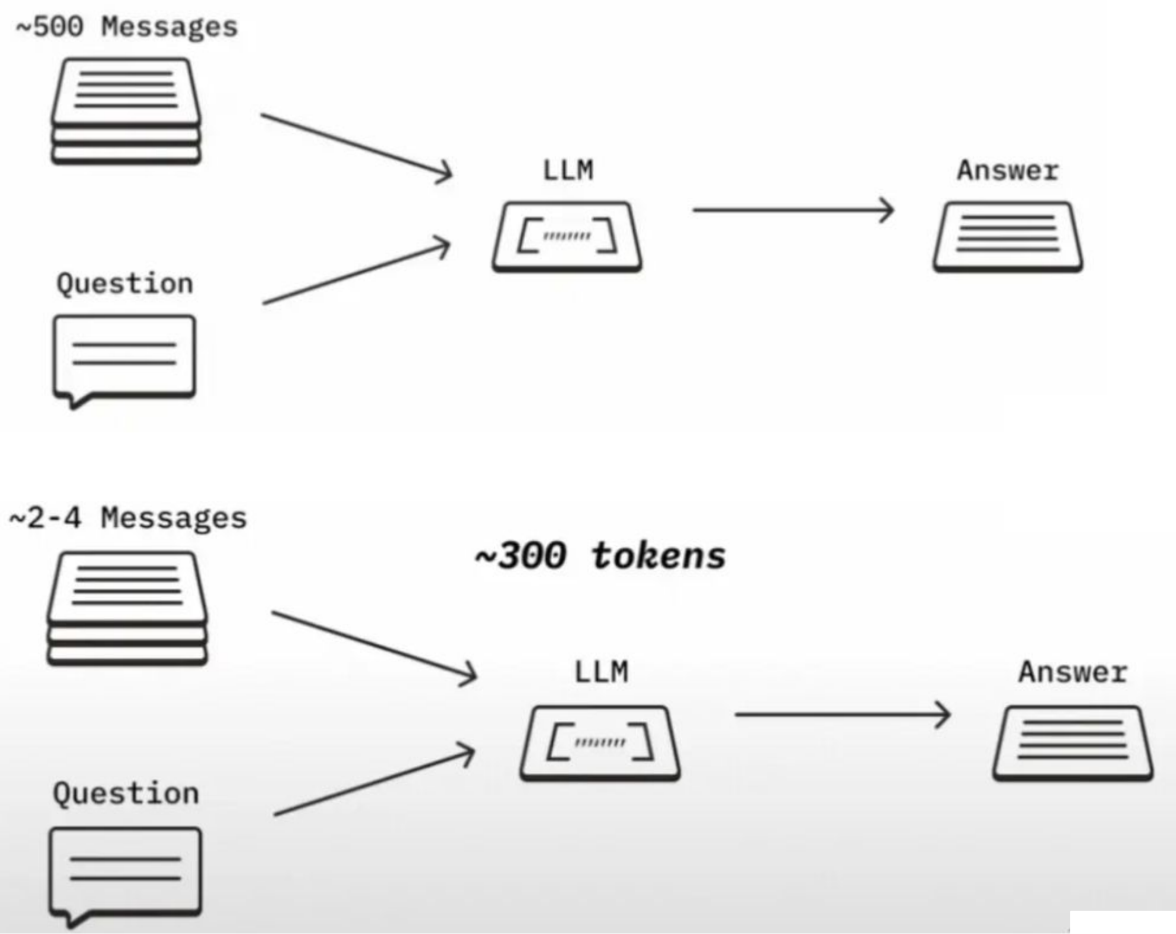
I recently moved to San Francisco, give me some restaurant recommendations

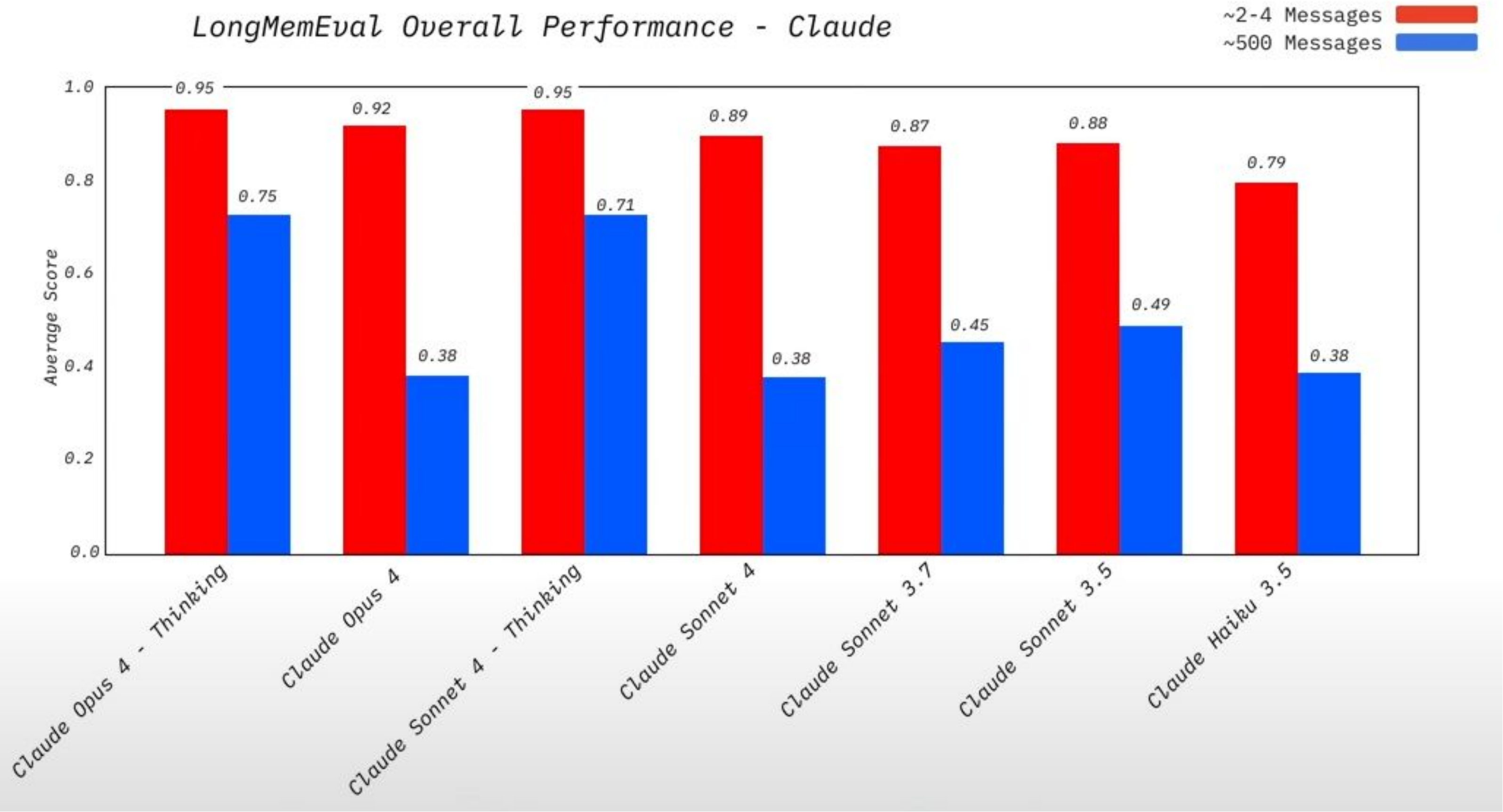
Here are some good restaurant recommendations for San Francisco...

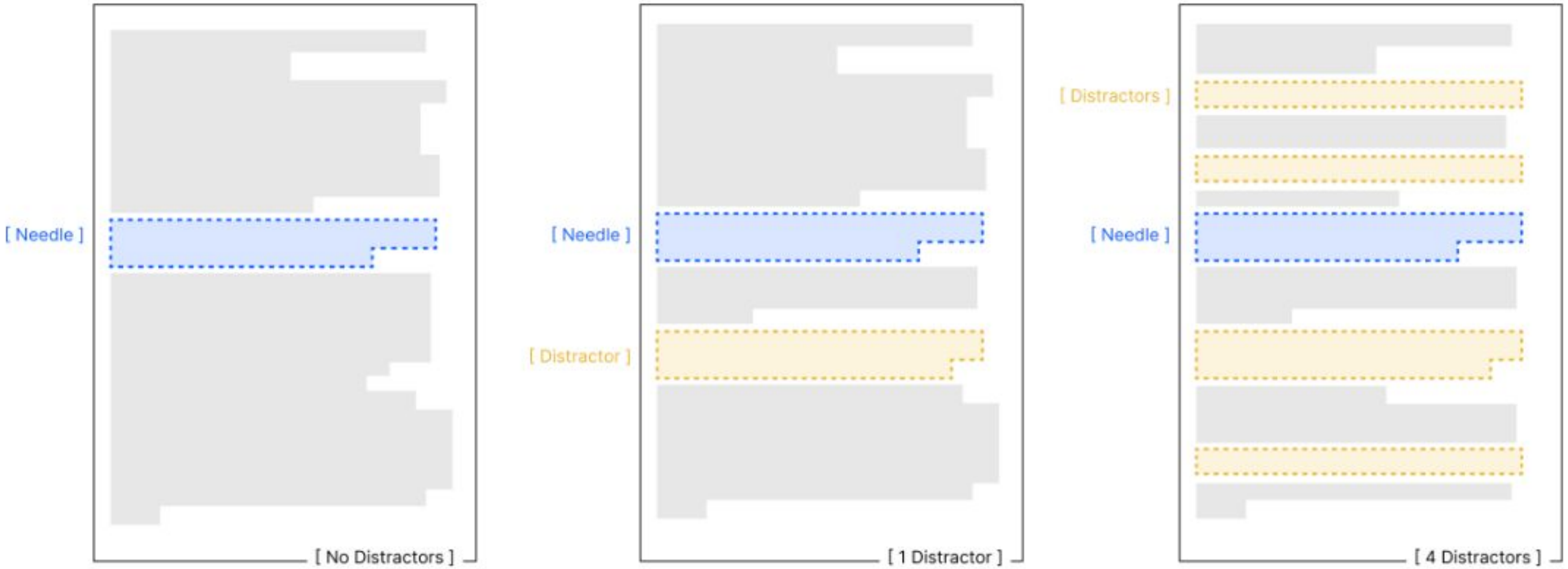
*What are some good outdoor activities to do on this beautiful sunny day?*

## Naive approach.


~500 Messages
Question
LLM
Answer


~500 Messages
Question
LLM
Answer
~2-4 Messages
~300 tokens
Question
LLM
Answer


LongMemEval Overall Performance - Claude
~2-4 Messages 🟥
~500 Messages 🟦
Average Score
1.0
0.8
0.6
0.4
0.2
0.0
0.95
0.75
0.92
0.38
0.95
0.71
0.89
0.38
0.87
0.45
0.88
0.49
0.79
0.38
Claude Opus 4 - Thinking
Claude Opus 4
Claude Sonnet 4 - Thinking
Claude Sonnet 4
Claude Sonnet 3.7
Claude Sonnet 3.5
Claude Haiku 3.5


[ Needle ]
[ No Distractors ]
[ Needle ]
[ Distractor ]
[ 1 Distractor ]
[ Distractors ]
[ Needle ]
[ 4 Distractors ]

# Distractors

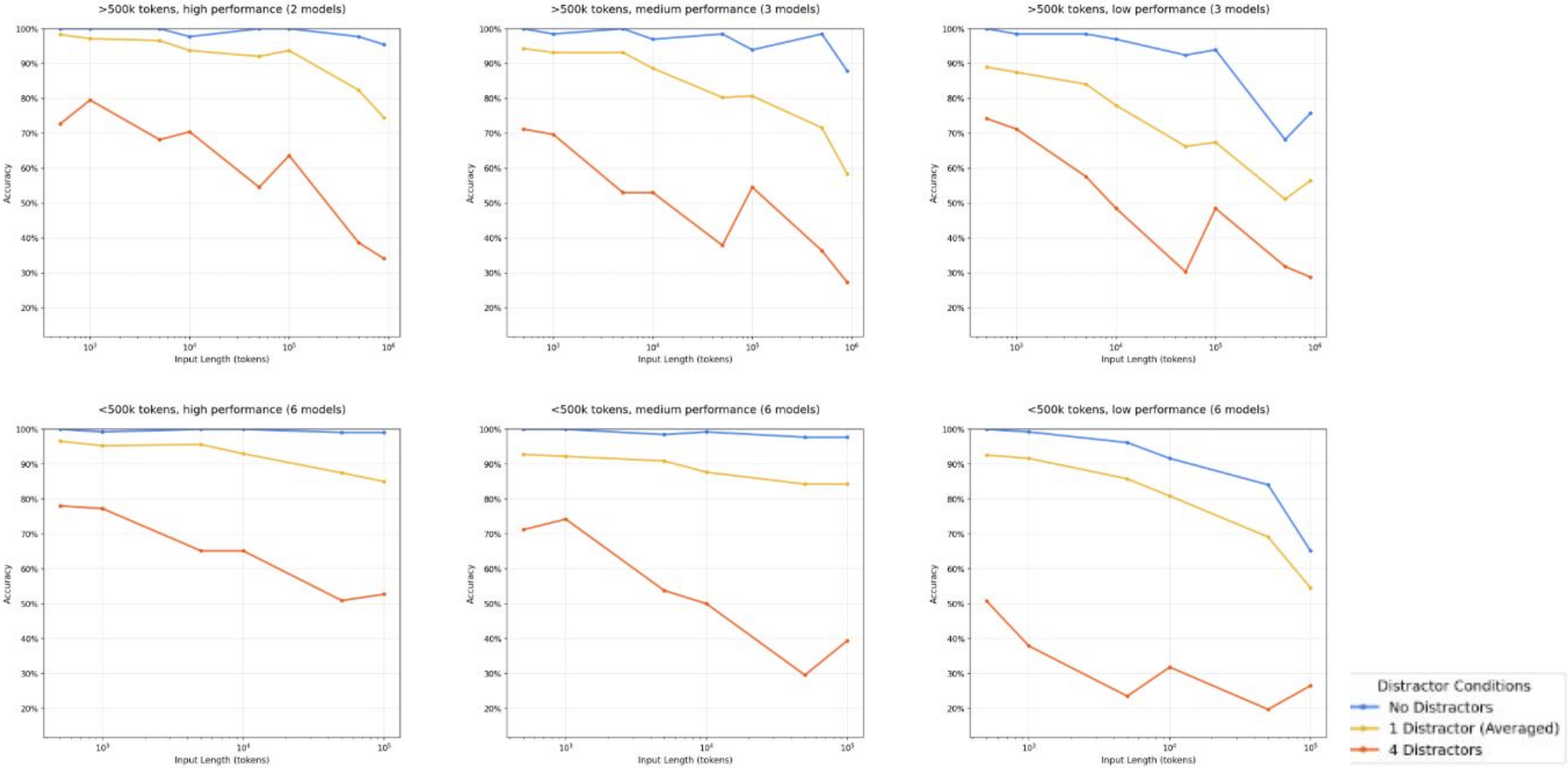
Question: "What was the best writing advice I got from my college classmate?"

Needle: "I think the best writing tip I received from my college classmate was to write every week."

Distractors:

- "The best writing tip I received from my college professor was to write everyday."
- "The worst writing advice I got from my college classmate was to write each essay in five different styles."
- "The best writing advice I got from my classmate was to write each essay in three different styles, this was back in high school."

# Distractors



# MCP

# Why MCP?

- Creates a universal language for AI Agents
- Allows networks of tools and models to work together
- Standardizes communication
- Adoption

# Working with LLMs

# Task performance is inconsistent

- Excellent at creative writing and drafting emails
- Inadequate for high precision tasks

What is

1234567 x 1234576

# Working with LLMs

- Demo
  - GPT-4-Turbo [ Chrome (incognito) ]

- After Demo
  - Why don't we hand off calculations to calculator?

# Using tools with LLMs

# Using tools with LLMs

- LLMs need to determine ***what*** needs to be done, not ***how*** to actually do it

Demo:

- OpenAI Assistant (without Code Interpreter)
- OpenAI Assistant + Code Interpreter
- It’s a sunny day

# Working with APIs

# Working with APIs

Demo:

- GPT-4-Turbo currency conversion

- API - Application Programming Interface
- APIs provide a standard way to access services
- APIs are everywhere.
- Need to give tools access to APIs

# Working with APIs

Demo:

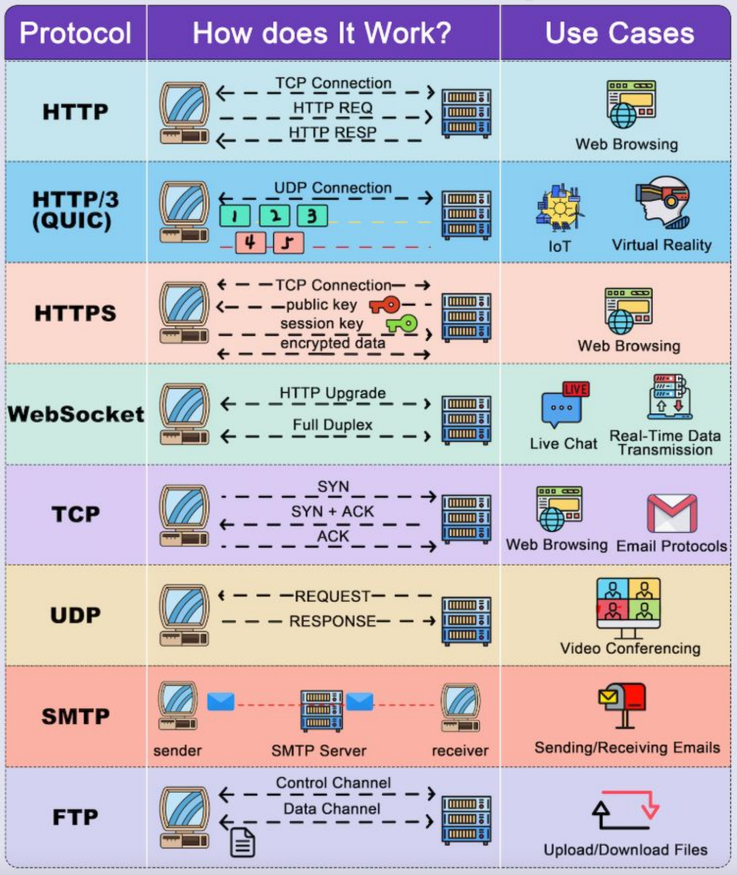
- GPT-4-Turbo (function) - Capture the Base & Target currency using 3-letter code and Amount
- Free text - Whatever they use in the UK
- https://www.exchangerate-api.com/ (jafdxc
  - Pair conversion
  -

# Working with APIs

- Tooling is not scaleable (multiplier, currency conversion)
- Require significant maintenance (v6)
- Security and privacy considerations (Bank accounts, health records)

# Using LLMs with MCP

# Using LLMs with MCP



| Protocol | How does It Work? | Use Cases |
|---|---|---|
| HTTP | TCP Connection<br>HTTP REQ<br>HTTP RESP | Web Browsing |
| HTTP/3 (QUIC) | UDP Connection<br>1 2 3<br>4 5 | IoT<br>Virtual Reality |
| HTTPS | TCP Connection<br>public key<br>session key<br>encrypted data | Web Browsing |
| WebSocket | HTTP Upgrade<br>Full Duplex | LIVE<br>Live Chat<br>Real-Time Data Transmission |
| TCP | SYN<br>SYN + ACK<br>ACK | Web Browsing<br>Email Protocols |
| UDP | REQUEST<br>RESPONSE | Video Conferencing |
| SMTP | sender<br>SMTP Server<br>receiver | Sending/Receiving Emails |
| FTP | Control Channel<br>Data Channel | Upload/Download Files |

# Using LLMs with MCP

https://modelcontextprotocol.io

# Host: The user-facing side

The Host’s job is to:

- Handle the user interface and permissions
- Kick off connections to MCP Servers via Clients
- Manage the back-and-forth between the user, the LLM, and any connected tools
- Present the final output clearly to the user

# Client: The Connector

The Client is a behind-the-scenes component inside the Host. It:

- Maintains a one-to-one connection with a Server
- Speaks the MCP protocol
- Translates requests from the Host into messages the Server understands

## Server: The Tool Provider

The Server is an external system that offers services or data to the AI model. It:

- Gives the model access to tools or databases
- Wraps existing functionality in a lightweight, AI-friendly interface
- Can be hosted locally or remotely
- Makes its features discoverable and usable through MCP

## Using LLMs with MCP

At its core, MCP is designed to embed your workflow into AI applications, giving essential context to any system using a large language model. This context could come in the form of tools or even just raw data.

# Using LLMs with MCP

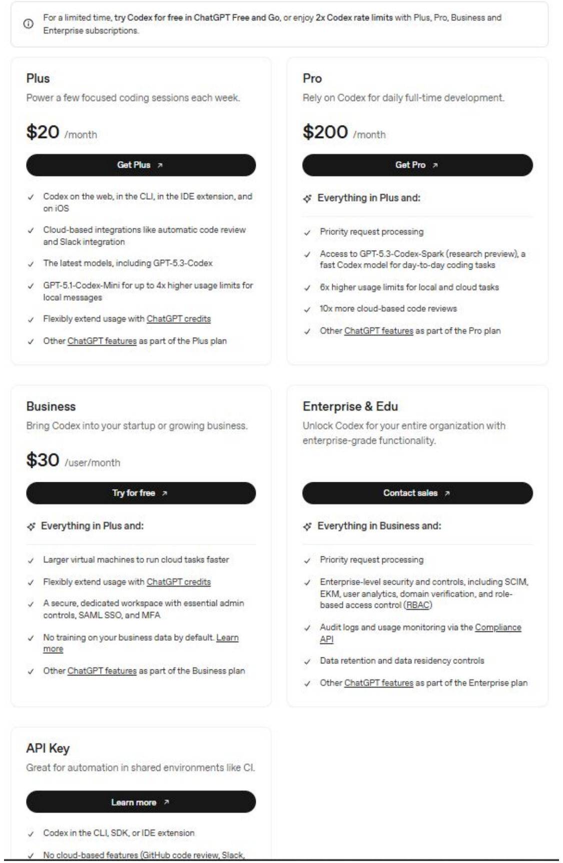
What do Large Language Models interact with?

They don't interact directly with APIs. They interact with prompts and tools and whatever you're giving the model to ingest.

# OpenAI codex lab

# Creating an app

- Required subscription
- Simple app (so creating it won't take long).
  - Create your own app. You don't have to use mine.
- Github account (and Github Pages)
- Basic knowledge of git



For a limited time, **try Codex for free in ChatGPT Free and Go**, or enjoy **2x Codex rate limits** with Plus, Pro, Business and Enterprise subscriptions.

**Plus**

Power a few focused coding sessions each week.

**$20** /month

- ✓ Codex on the web, in the CLI, in the IDE extension, and on iOS
- ✓ Cloud-based integrations like automatic code review and Slack integration
- ✓ The latest models, including GPT-5.3-Codex
- ✓ GPT-5.1-Codex-Mini for up to 4x higher usage limits for local messages
- ✓ Flexibly extend usage with ChatGPT credits
- ✓ Other ChatGPT features as part of the Plus plan

**Pro**

Rely on Codex for daily full-time development.

**$200** /month

Everything in Plus and:

- ✓ Priority request processing
- ✓ Access to GPT-5.3-Codex-Spark (research preview), a fast Codex model for day-to-day coding tasks
- ✓ 6x higher usage limits for local and cloud tasks
- ✓ 10x more cloud-based code reviews
- ✓ Other ChatGPT features as part of the Pro plan

**Business**

Bring Codex into your startup or growing business.

**$30** /user/month

Everything in Plus and:

- ✓ Larger virtual machines to run cloud tasks faster
- ✓ Flexibly extend usage with ChatGPT credits
- ✓ A secure, dedicated workspace with essential admin controls, SAML SSO, and MFA
- ✓ No training on your business data by default. Learn more
- ✓ Other ChatGPT features as part of the Business plan

**Enterprise & Edu**

Unlock Codex for your entire organization with enterprise-grade functionality.

Everything in Business and:

- ✓ Priority request processing
- ✓ Enterprise-level security and controls, including SCIM, EKM, user analytics, domain verification, and role-based access control (RBAC)
- ✓ Audit logs and usage monitoring via the Compliance API
- ✓ Data retention and data residency controls
- ✓ Other ChatGPT features as part of the Enterprise plan

**API Key**

Great for automation in shared environments like CI.

- ✓ Codex in the CLI, SDK, or IDE extension
- ✓ No cloud-based features (GitHub code review, Slack,

# App

INITIAL

- Demo working example at /life

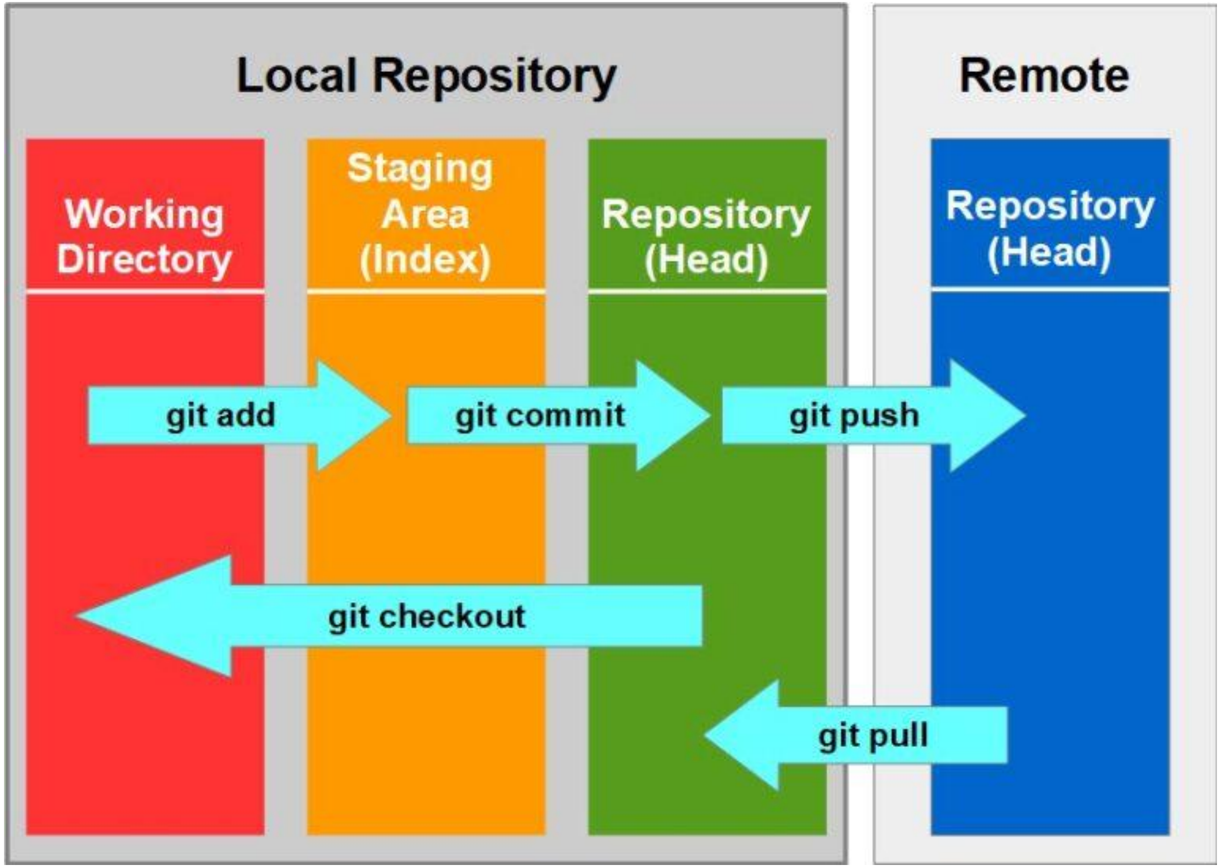
- /dot/app.py uses the Python flash framework to create an interactive webpage
- GitHub Pages doesn’t support flash
- Using /dot/app.py file create an equivalent css/html page

OPTIONAL

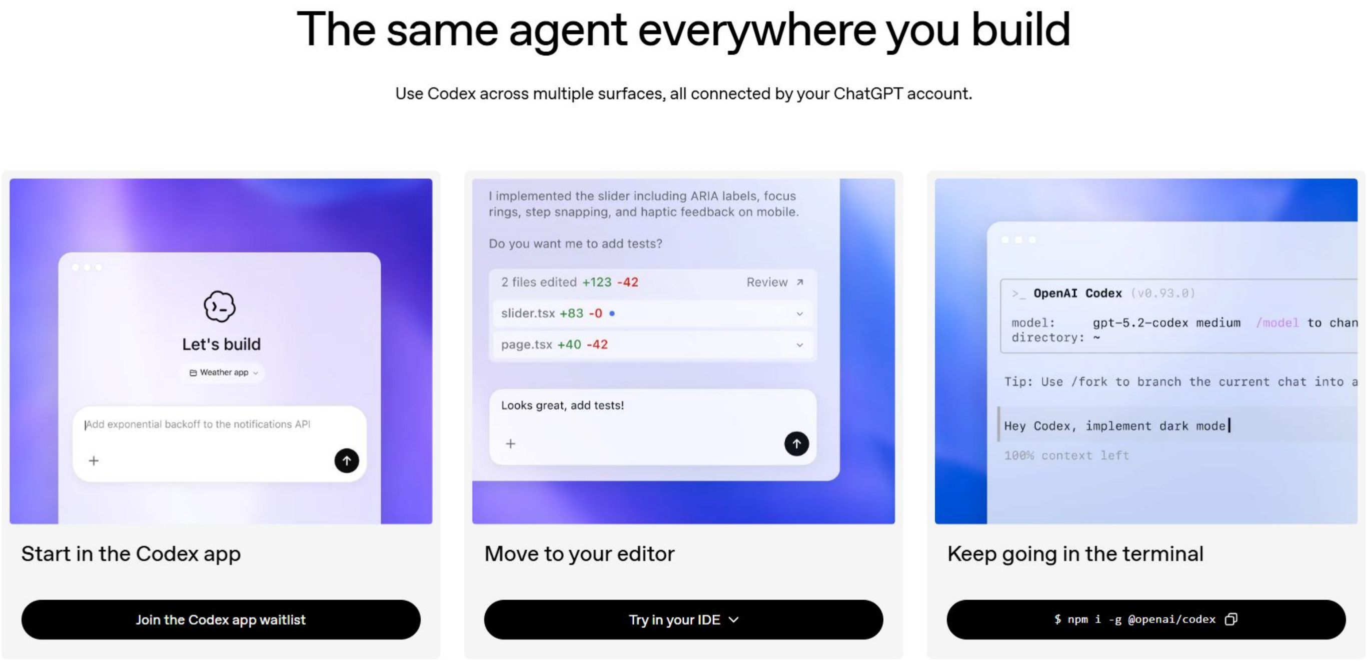
- Display images a screenful at a time
- Support dark mode

What codex needs to know
The file test-dot/app.py uses flask. Change it so that I have the same look and feel using css and html. I want the final files to run on jonfernandes.github.io/dot. Push the changes to Github

# Git basics


Local Repository
Remote
Working Directory
Staging Area (Index)
Repository (Head)
Repository (Head)
git add
git commit
git push
git checkout
git pull

# Codex



## The same agent everywhere you build

Use Codex across multiple surfaces, all connected by your ChatGPT account.

### Start in the Codex app

I implemented the slider including ARIA labels, focus rings, step snapping, and haptic feedback on mobile.

Do you want me to add tests?

Looks great, add tests!

### Move to your editor

```
>_ OpenAI Codex (v0.93.0)

model:     gpt-5.2-codex medium   /model to chan
directory: ~

Tip: Use /fork to branch the current chat into a

Hey Codex, implement dark mode
100% context left
```

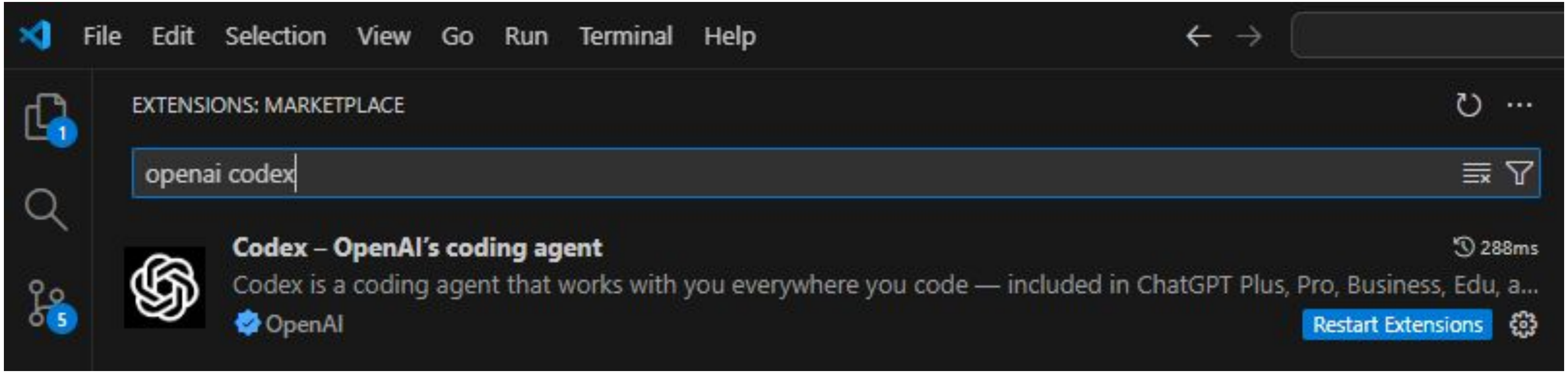
### Keep going in the terminal

```
$ npm i -g @openai/codex
```

# Codex app

# Code editor

# Using code editor



openai codex

**Codex – OpenAI's coding agent**

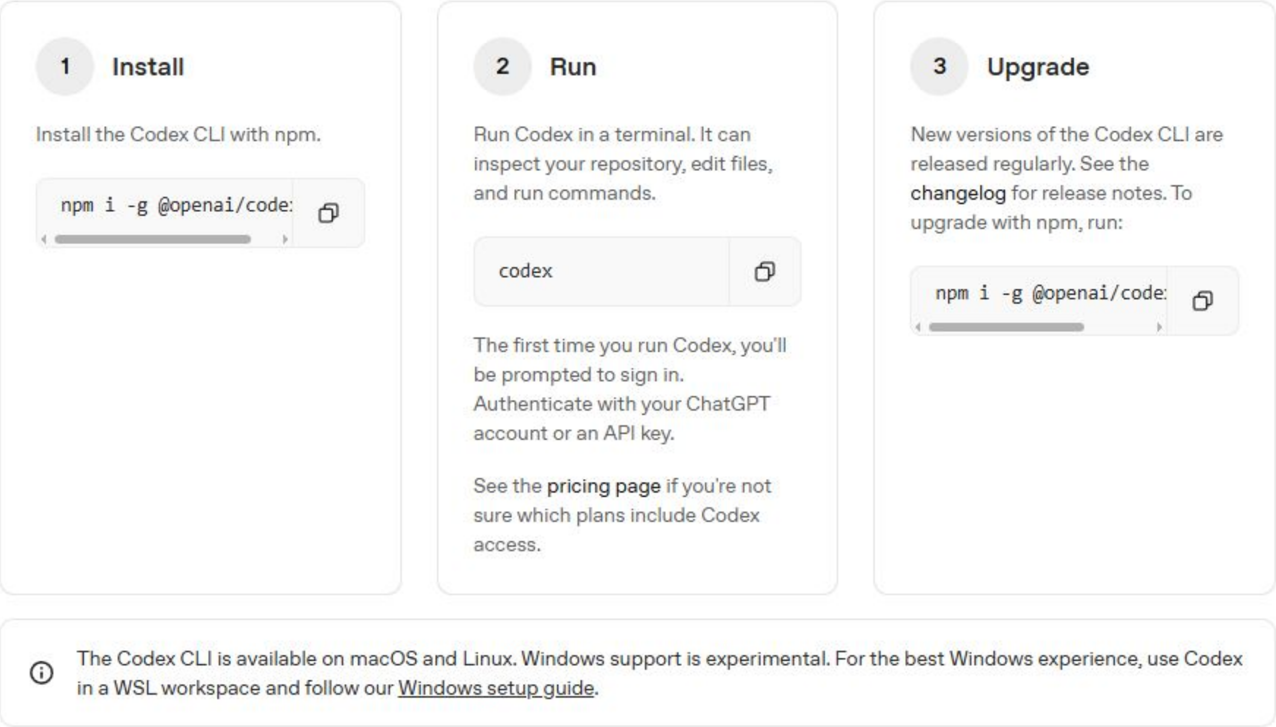
288ms

Codex is a coding agent that works with you everywhere you code — included in ChatGPT Plus, Pro, Business, Edu, a...

OpenAI

# CLI – Command Line Interface

# Codex CLI



### 1 Install

Install the Codex CLI with npm.

```
npm i -g @openai/code
```

### 2 Run

Run Codex in a terminal. It can inspect your repository, edit files, and run commands.

```
codex
```

The first time you run Codex, you'll be prompted to sign in. Authenticate with your ChatGPT account or an API key.

See the **pricing page** if you're not sure which plans include Codex access.

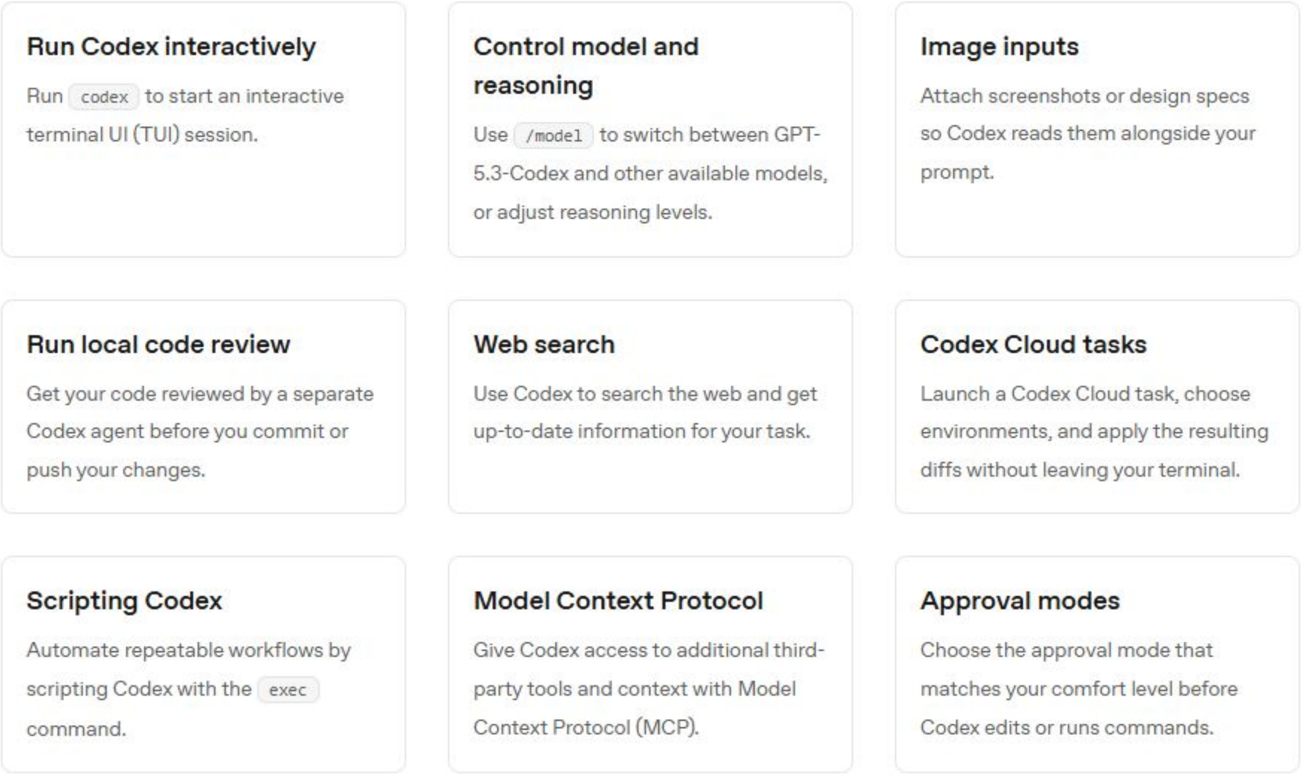
### 3 Upgrade

New versions of the Codex CLI are released regularly. See the **changelog** for release notes. To upgrade with npm, run:

```
npm i -g @openai/code
```

ⓘ The Codex CLI is available on macOS and Linux. Windows support is experimental. For the best Windows experience, use Codex in a WSL workspace and follow our Windows setup guide.

# Working with Codex CLI



### Run Codex interactively

Run `codex` to start an interactive terminal UI (TUI) session.

### Control model and reasoning

Use `/model` to switch between GPT-5.3-Codex and other available models, or adjust reasoning levels.

### Image inputs

Attach screenshots or design specs so Codex reads them alongside your prompt.

### Run local code review

Get your code reviewed by a separate Codex agent before you commit or push your changes.

### Web search

Use Codex to search the web and get up-to-date information for your task.

### Codex Cloud tasks

Launch a Codex Cloud task, choose environments, and apply the resulting diffs without leaving your terminal.

### Scripting Codex

Automate repeatable workflows by scripting Codex with the `exec` command.

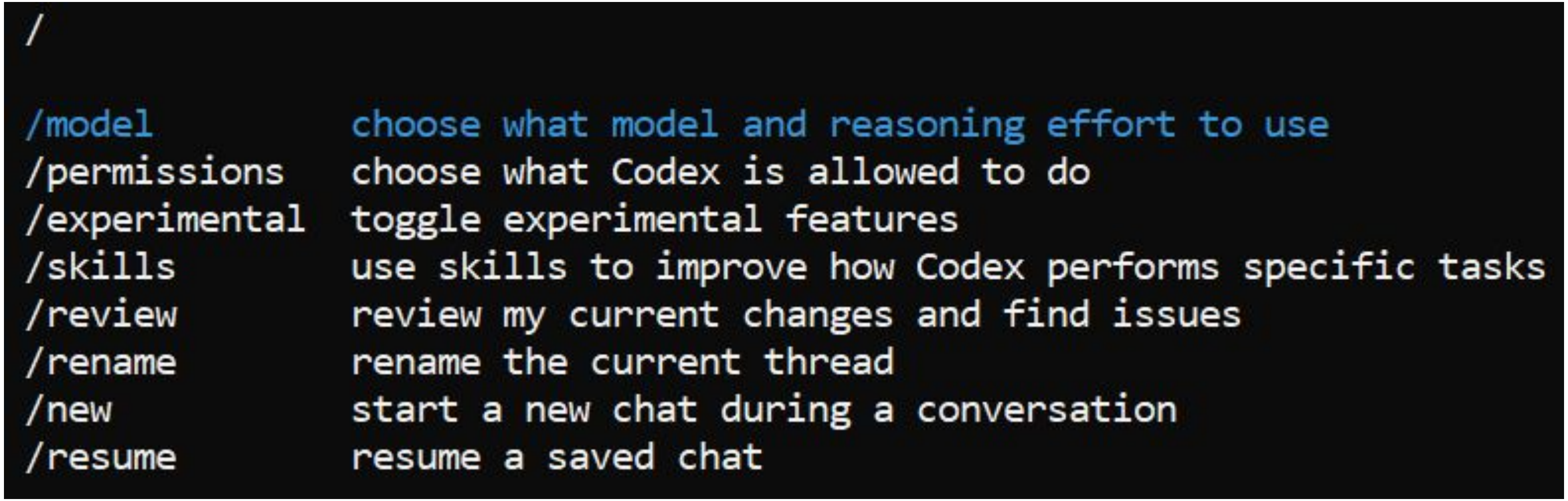
### Model Context Protocol

Give Codex access to additional third-party tools and context with Model Context Protocol (MCP).

### Approval modes

Choose the approval mode that matches your comfort level before Codex edits or runs commands.

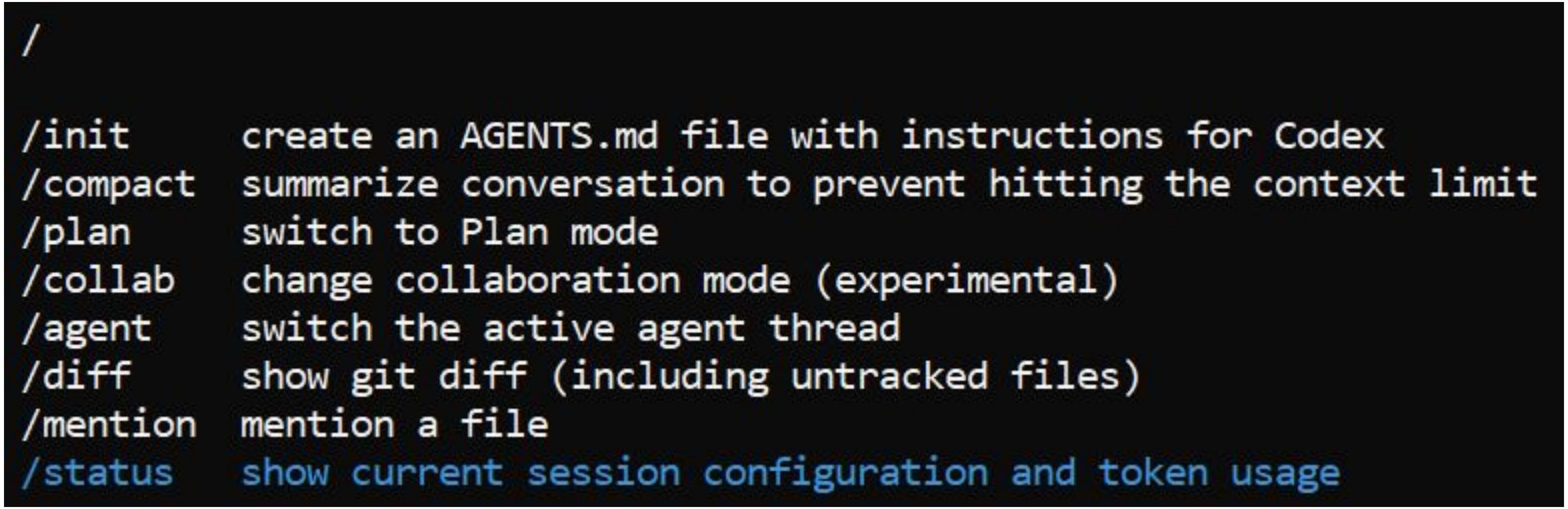
# Working with Codex CLI



```
/

/model          choose what model and reasoning effort to use
/permissions    choose what Codex is allowed to do
/experimental   toggle experimental features
/skills         use skills to improve how Codex performs specific tasks
/review         review my current changes and find issues
/rename         rename the current thread
/new            start a new chat during a conversation
/resume         resume a saved chat
```

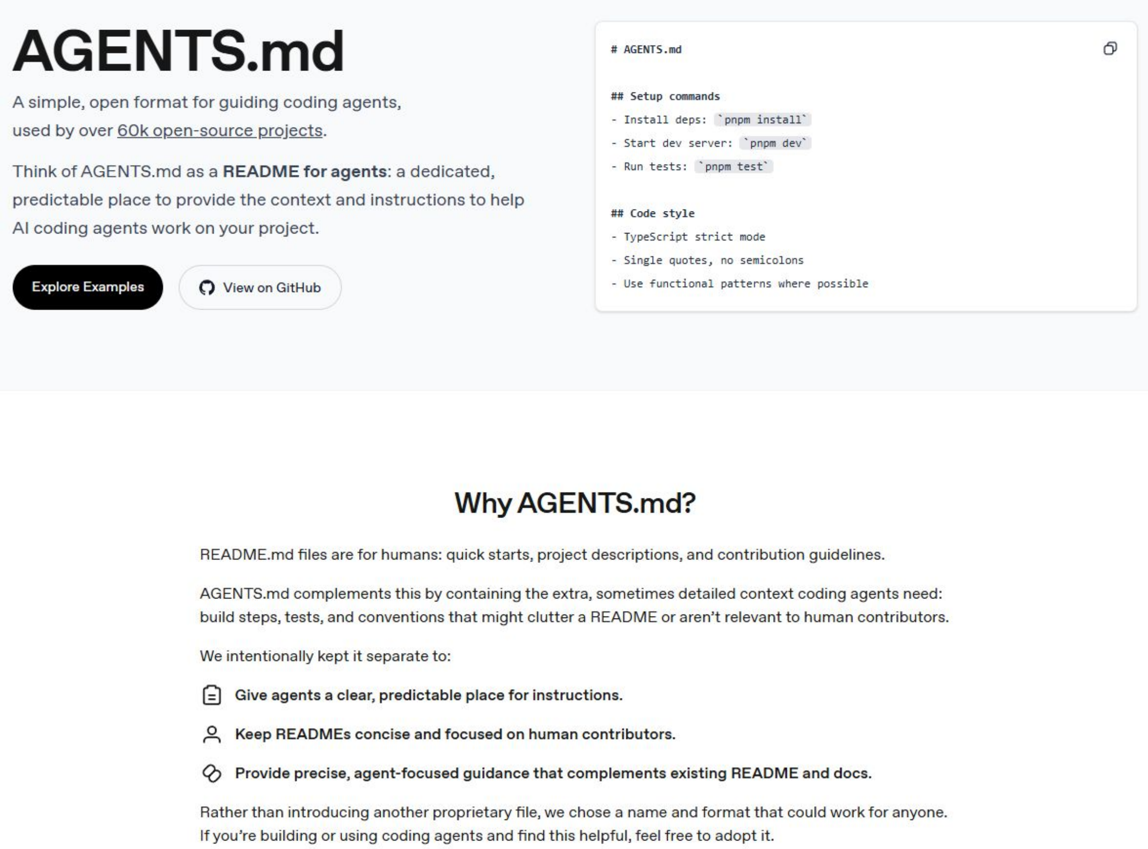
# Working with Codex CLI



```
/

/init      create an AGENTS.md file with instructions for Codex
/compact   summarize conversation to prevent hitting the context limit
/plan      switch to Plan mode
/collab    change collaboration mode (experimental)
/agent     switch the active agent thread
/diff      show git diff (including untracked files)
/mention   mention a file
/status    show current session configuration and token usage
```

# AGENTS.md using /init



## AGENTS.md

A simple, open format for guiding coding agents, used by over 60k open-source projects.

Think of AGENTS.md as a **README for agents**: a dedicated, predictable place to provide the context and instructions to help AI coding agents work on your project.

```
# AGENTS.md

## Setup commands
- Install deps: `pnpm install`
- Start dev server: `pnpm dev`
- Run tests: `pnpm test`

## Code style
- TypeScript strict mode
- Single quotes, no semicolons
- Use functional patterns where possible
```

### Why AGENTS.md?

README.md files are for humans: quick starts, project descriptions, and contribution guidelines.

AGENTS.md complements this by containing the extra, sometimes detailed context coding agents need: build steps, tests, and conventions that might clutter a README or aren't relevant to human contributors.

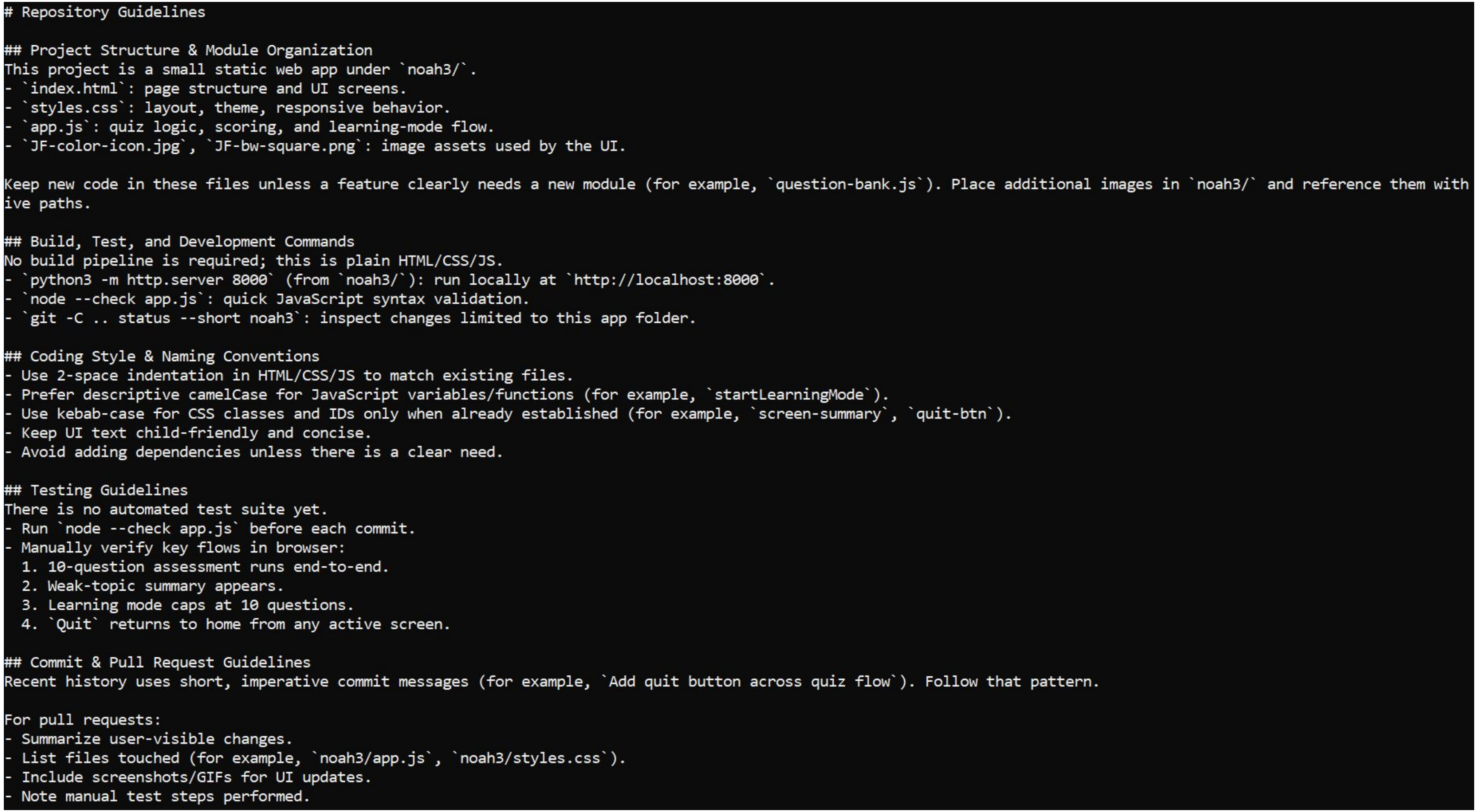
We intentionally kept it separate to:

- **Give agents a clear, predictable place for instructions.**
- **Keep READMEs concise and focused on human contributors.**
- **Provide precise, agent-focused guidance that complements existing README and docs.**

Rather than introducing another proprietary file, we chose a name and format that could work for anyone. If you're building or using coding agents and find this helpful, feel free to adopt it.

https://agents.md

# AGENTS.md



```
# Repository Guidelines

## Project Structure & Module Organization
This project is a small static web app under `noah3/`.
- `index.html`: page structure and UI screens.
- `styles.css`: layout, theme, responsive behavior.
- `app.js`: quiz logic, scoring, and learning-mode flow.
- `JF-color-icon.jpg`, `JF-bw-square.png`: image assets used by the UI.

Keep new code in these files unless a feature clearly needs a new module (for example, `question-bank.js`). Place additional images in `noah3/` and reference them with
ive paths.

## Build, Test, and Development Commands
No build pipeline is required; this is plain HTML/CSS/JS.
- `python3 -m http.server 8000` (from `noah3/`): run locally at `http://localhost:8000`.
- `node --check app.js`: quick JavaScript syntax validation.
- `git -C .. status --short noah3`: inspect changes limited to this app folder.

## Coding Style & Naming Conventions
- Use 2-space indentation in HTML/CSS/JS to match existing files.
- Prefer descriptive camelCase for JavaScript variables/functions (for example, `startLearningMode`).
- Use kebab-case for CSS classes and IDs only when already established (for example, `screen-summary`, `quit-btn`).
- Keep UI text child-friendly and concise.
- Avoid adding dependencies unless there is a clear need.

## Testing Guidelines
There is no automated test suite yet.
- Run `node --check app.js` before each commit.
- Manually verify key flows in browser:
  1. 10-question assessment runs end-to-end.
  2. Weak-topic summary appears.
  3. Learning mode caps at 10 questions.
  4. `Quit` returns to home from any active screen.

## Commit & Pull Request Guidelines
Recent history uses short, imperative commit messages (for example, `Add quit button across quiz flow`). Follow that pattern.

For pull requests:
- Summarize user-visible changes.
- List files touched (for example, `noah3/app.js`, `noah3/styles.css`).
- Include screenshots/GIFs for UI updates.
- Note manual test steps performed.
```

# Fine-tuning

# Why fine-tuning?

- Improve task performance
- Adapt to a specific domain
- Enforce a certain output style
- Lower inference cost and latency

# Fine-tuning Lab

https://colab.research.google.com/drive/1qLgiL0kQ008PX5NCR29cv3T7CmmL9e_y?usp=sharing

O'REILLY®